

EECE 5639 Computer Vision I

Lecture 22

Object Recognition: Eigenimages, Bag of Words

Project 4 is out. Now Due April 16.

Hw 5 is out. Now Due April 19

Next Class

Object Recognition: deep learning

PCA Theorem

Let $x_1 x_2 \dots x_n$ be a set of $n N^2 \times 1$ vectors and let \bar{x} be their average:

$$x_i = \begin{bmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{iN^2} \end{bmatrix} \quad \bar{x} = \frac{1}{n} \sum_{i=1}^{i=n} \begin{bmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{iN^2} \end{bmatrix}$$

Note: Each $N \times N$ image template can be represented as a $N^2 \times 1$ vector whose elements are the template pixel values.

PCA Theorem

Let X be the $N^2 \times n$ matrix:

$$X = \begin{bmatrix} \mathbf{x}_1 - \bar{\mathbf{x}} & \mathbf{x}_2 - \bar{\mathbf{x}} & \cdots & \mathbf{x}_n - \bar{\mathbf{x}} \end{bmatrix}$$

Note: subtracting the mean is equivalent to translating the coordinate system to the location of the mean.

PCA Theorem

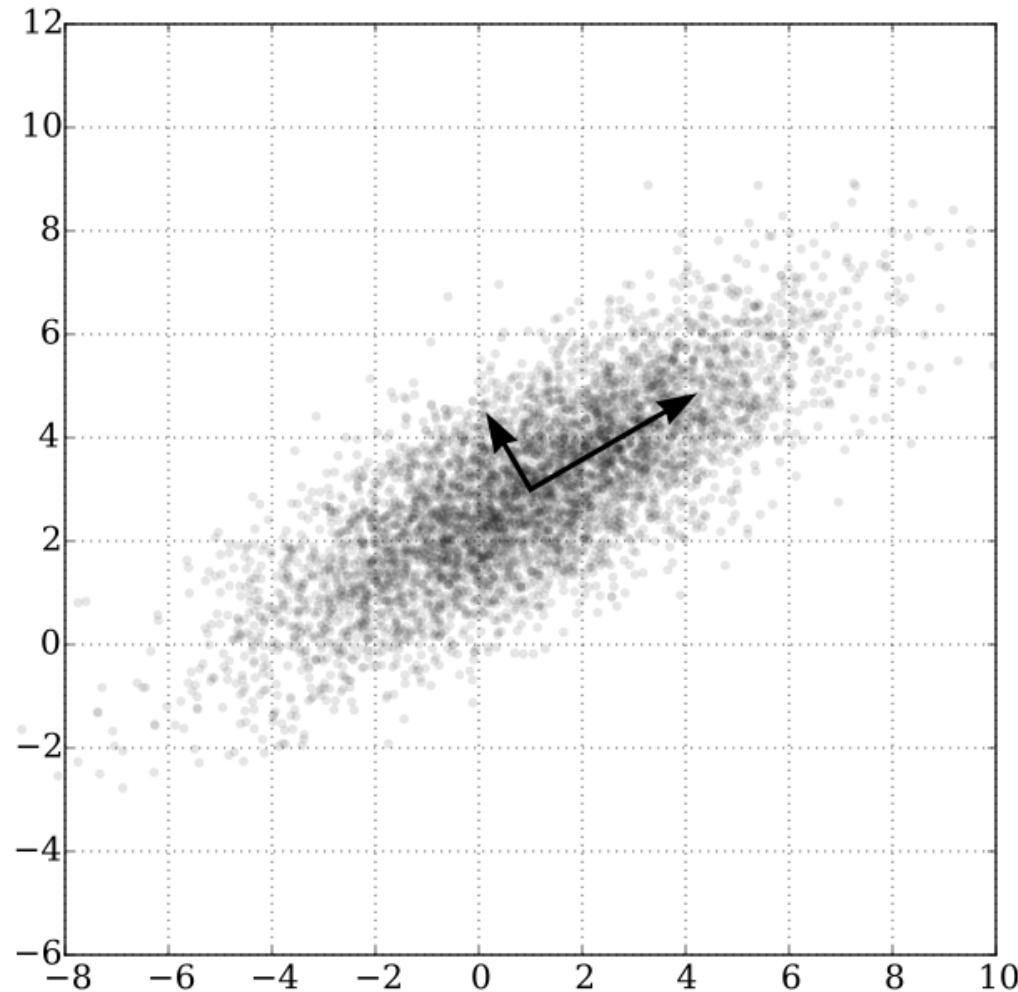
Let $Q = X X^T$ be the $N^2 \times N^2$ matrix:

$$Q = X X^T = \begin{bmatrix} \mathbf{x}_1 - \bar{\mathbf{x}} & \mathbf{x}_2 - \bar{\mathbf{x}} & \cdots & \mathbf{x}_n - \bar{\mathbf{x}} \end{bmatrix} \begin{bmatrix} (\mathbf{x}_1 - \bar{\mathbf{x}})^T \\ (\mathbf{x}_2 - \bar{\mathbf{x}})^T \\ \vdots \\ (\mathbf{x}_n - \bar{\mathbf{x}})^T \end{bmatrix}$$

Notes:

1. Q is square
2. Q is symmetric
3. Q is the covariance matrix
4. Q can be very large (remember that N^2 is the number of pixels in the template)

PCA



PCA Theorem

Theorem:

Each x_j can be written as: $x_j = \bar{x} + \sum_{i=1}^{i=n} g_{ji} e_i$

where e_i are the n eigenvectors of Q with non-zero eigenvalues.

Notes:

1. The eigenvectors $e_1 e_2 \dots e_n$ span an **eigenspace**
2. $e_1 e_2 \dots e_n$ are $N^2 \times 1$ orthonormal vectors ($N \times N$ images).
3. The scalars g_{ji} are the coordinates of x_j in the space.
4. $g_{ji} = (x_j - \bar{x}) \cdot e_i$

Using PCA to Compress Data

Sort the eigenvectors e_i according to their eigenvalue:

$$\lambda_1 \geq \lambda_2 \geq \dots \lambda_n$$

- Assuming that $\lambda_i \approx 0$ if $i > k$

- Then

$$\mathbf{x}_j \approx \bar{\mathbf{x}} + \sum_{i=1}^{i=k} g_{ji} \mathbf{e}_i$$

Eigenspaces: Efficient Image Storage



- Use PCA to compress the data:
 - each image is stored as a k-dimensional vector
 - Need to store $k N \times N$ eigenvectors
 - $k \ll n \ll N^2$

$$\begin{matrix} \text{[Truck Image]} & \cong & \text{[Truck Image]} & = a_{01} & \text{[Eigenvector]} & + a_{02} & \text{[Eigenvector]} & + a_{03} & \text{[Eigenvector]} & + a_{04} & \text{[Eigenvector]} & + a_{05} & \text{[Eigenvector]} & + a_{06} & \text{[Eigenvector]} & + \dots \end{matrix}$$

Eigenspaces: Efficient Image Comparison



- Use the same procedure to compress the given image to a k -dimensional vector.
- Compare the compressed vectors:
 - Dot product of k -dimensional vectors
 - $k \ll n \ll N^2$

$$\begin{matrix} \text{[Truck Image]} & \cong & \text{[Truck Image]} & = a_{01} & \text{[Eigenbasis Image]} & + a_{02} & \text{[Eigenbasis Image]} & + a_{03} & \text{[Eigenbasis Image]} & + a_{04} & \text{[Eigenbasis Image]} & + a_{05} & \text{[Eigenbasis Image]} & + a_{06} & \text{[Eigenbasis Image]} & + \dots \end{matrix}$$

Implementing PCA

Need to find “first” k eigenvectors of Q:

$$Q = XX^T = \begin{bmatrix} \mathbf{x}_1 - \bar{\mathbf{x}} & \mathbf{x}_2 - \bar{\mathbf{x}} & \cdots & \mathbf{x}_n - \bar{\mathbf{x}} \end{bmatrix} \begin{bmatrix} (\mathbf{x}_1 - \bar{\mathbf{x}})^T \\ (\mathbf{x}_2 - \bar{\mathbf{x}})^T \\ \vdots \\ (\mathbf{x}_n - \bar{\mathbf{x}})^T \end{bmatrix}$$

Q is $N^2 \times N^2$ where N^2 is the number of pixels in each image.
For a 256 x 256 image, $N^2 = 65536 !!$

Finding ev of Q

$Q=XX^T$ is very large. Instead, consider the matrix $P=X^TX$

- Q and P are both symmetric, but $Q \neq P^T$
- Q is $N^2 \times N^2$, P is $n \times n$
- n is the number of training images, typically $n \ll N$

Finding ev of Q

Let e be an eigenvector of P with eigenvalue λ :

$$Pe = \lambda e$$

$$X^T X e = \lambda e$$

$$XX^T X e = \lambda X e$$

$$Q(Xe) = \lambda(Xe)$$

Xe is an eigenvector of Q also with eigenvalue λ !

Singular Value Decomposition (SVD)

Any $m \times n$ matrix X can be written as the product of 3 matrices:

$$X = UDV^T$$

Where:

- U is $m \times m$ and its columns are orthonormal vectors
- V is $n \times n$ and its columns are orthonormal vectors
- D is $m \times n$ diagonal and its diagonal elements are called the singular values of X , and are such that:

$$\sigma_1, \sigma_2, \dots, \sigma_n >= 0$$

SVD Properties

$$X = UDV^T$$

- The columns of U are the eigenvectors of $Q = XX^T$
- The columns of V are the eigenvectors of $P = X^TX$
- The squares of the diagonal elements of D are the eigenvalues of XX^T and X^TX

Algorithm EIGENSPACE_LEARN

Assumptions:

1. Each image contains one object only.
2. Objects are imaged by a fixed camera .
3. Images are normalized in size N x N:
The image frame is the minimum rectangle enclosing the object.
4. Energy of pixels values is normalized to 1:
 $\sum_i \sum_j I(i,j)^2 = 1$
5. The object is completely visible and unoccluded in all images.

Algorithm EIGENSPACE_LEARN

Getting the data:

For each object o to be represented, $o = 1, \dots, O$

1. Place o on a turntable, acquire a set of n images by rotating the table in increments of $360^\circ/n$
2. For each image p , $p = 1, \dots, n$:
 1. Segment o from the background
 2. Normalize the image size and energy
 3. Arrange the pixels as vectors \mathbf{x}_p^o

Algorithm EIGENSPACE_LEARN

Storing the data:

1. Find the average image vector $\bar{\mathbf{x}} = \frac{1}{n.o} \sum_{o=1}^O \sum_{p=1}^n x_p^o$

2. Assemble the matrix X:

$$X = [\mathbf{x}_1^1 - \bar{\mathbf{x}} \ \ \mathbf{x}_2^1 - \bar{\mathbf{x}} \ \ \dots \ \ \mathbf{x}_n^o - \bar{\mathbf{x}}]$$

3. Find the first k eigenvectors of XX^T : e_1, \dots, e_k
(use X^TX or SVD)

4. For each object o, each image p:

• Compute the corresponding k-dimensional point:

$$\mathbf{g}_p^o = [e_1 \ e_2 \ \dots \ e_k] (\mathbf{x}_p^o - \bar{\mathbf{x}})$$

Algorithm EIGENSPACE_IDENTIF

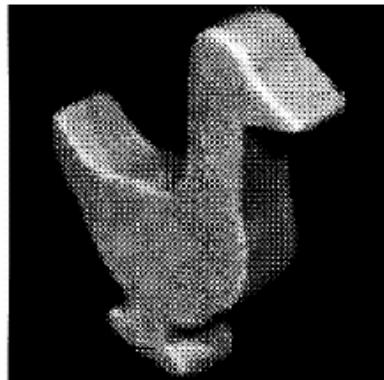
Recognizing an object from the DB:

1. Given an image, segment the object from the background
2. Normalize the size an energy, write it as a vector \mathbf{i}
3. Compute the corresponding k-dimensional point:

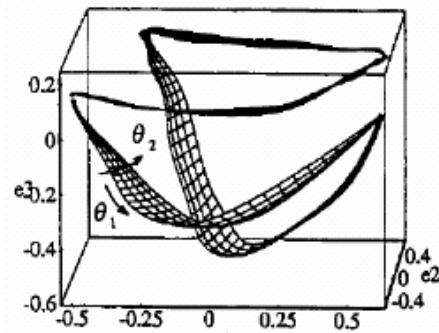
$$\mathbf{g} = [\mathbf{e}_1 \ \mathbf{e}_2 \ \cdots \ \mathbf{e}_k] (\mathbf{i} - \bar{\mathbf{x}})$$

4. Find the closest \mathbf{g}^o_p k-dimensional point to \mathbf{g}

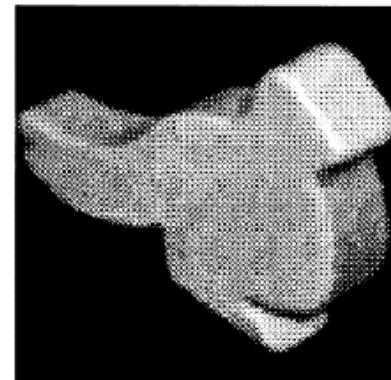
Appearance Manifolds



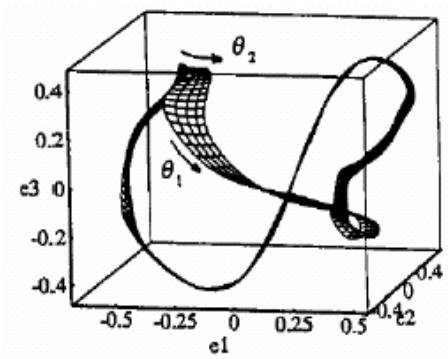
A



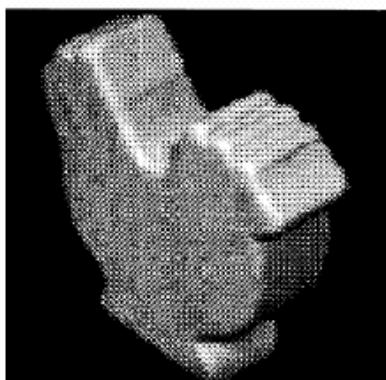
A



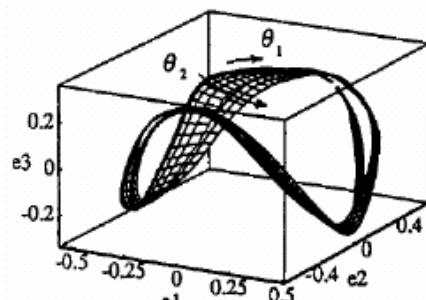
B



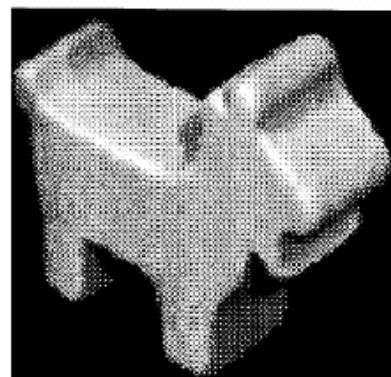
B



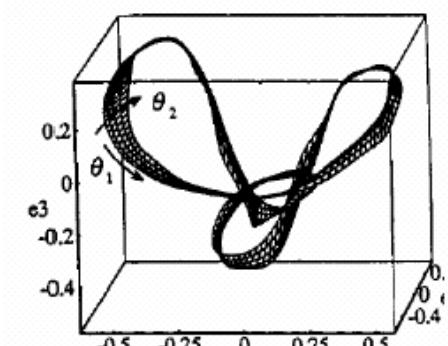
C



C



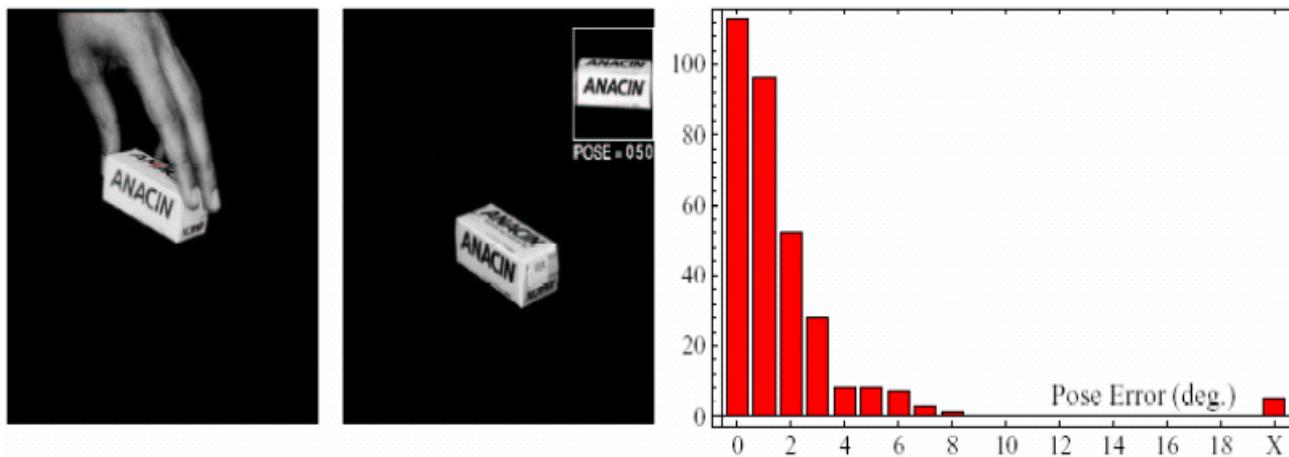
D



D

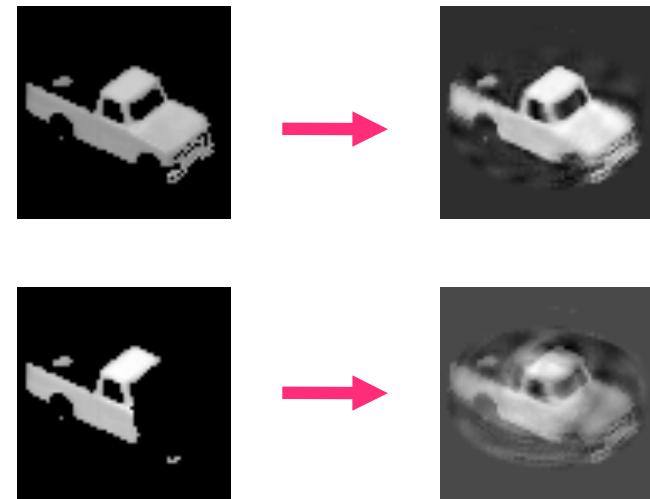
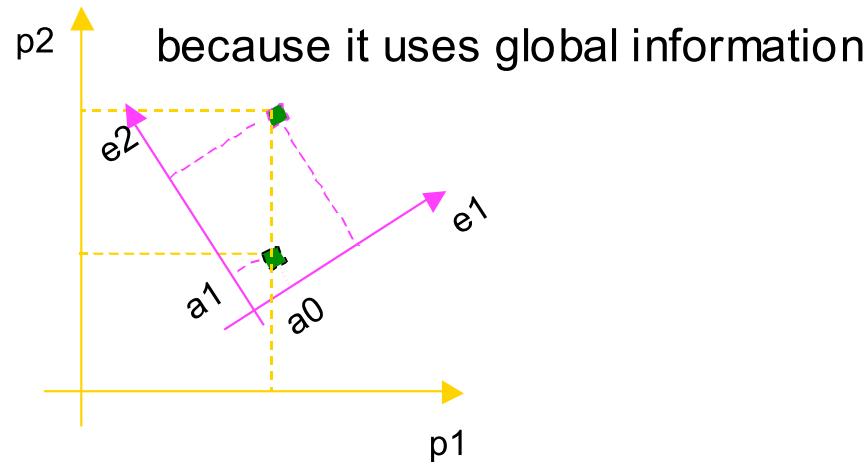
[Murase & Nayar, 1996]

Appearance Manifolds



Murase & Nayar

PCA has problems with occlusion



Example 3: ABPs and ABRs

Use local appearance:

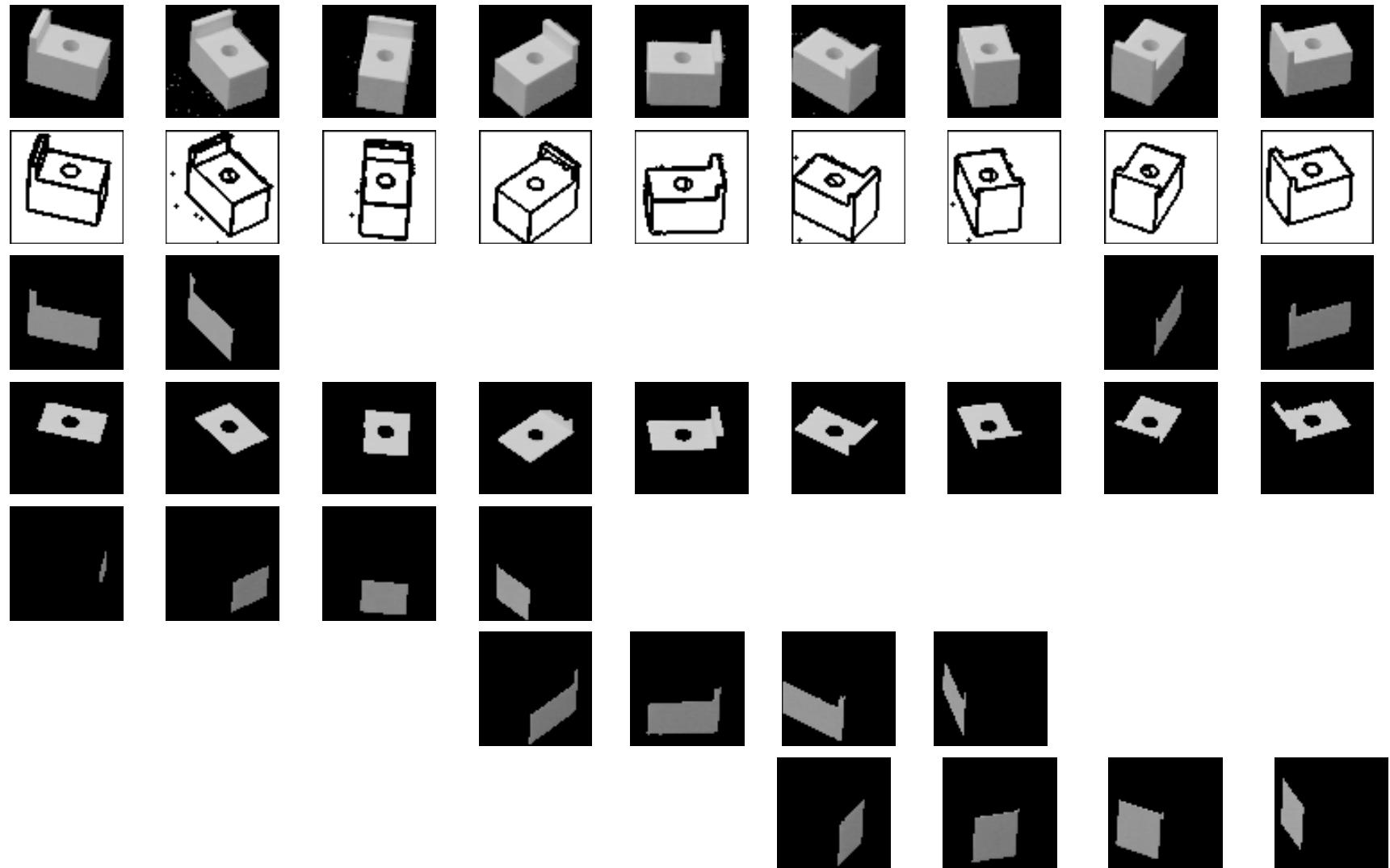
PARTS and RELATIONS

- Can model general-form objects
- Can deal with occlusion and clutter
- Can handle translation and scaling
- It is robust to segmentation problems

Parts from Images

PARTS are image regions segmented using some segmentation algorithm.

Appearance-Based Parts



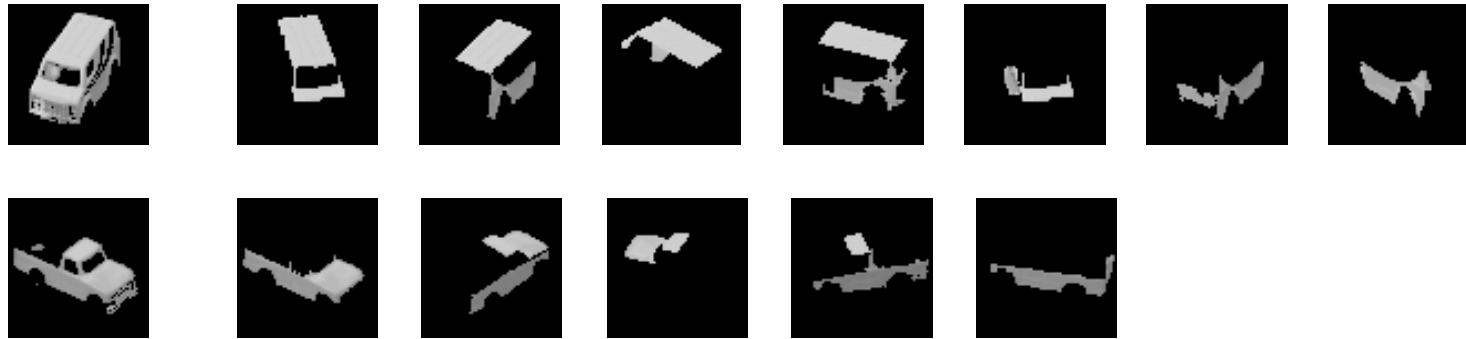
Appearance-Based Relations

Several objects can have similar parts

Use spatial relations to discriminate

Represent relations using PCA

Examples of ABRs



Recognition in cluttered scenes

Segment the given image

PARTS:

- Project regions into the ABP eigenspace

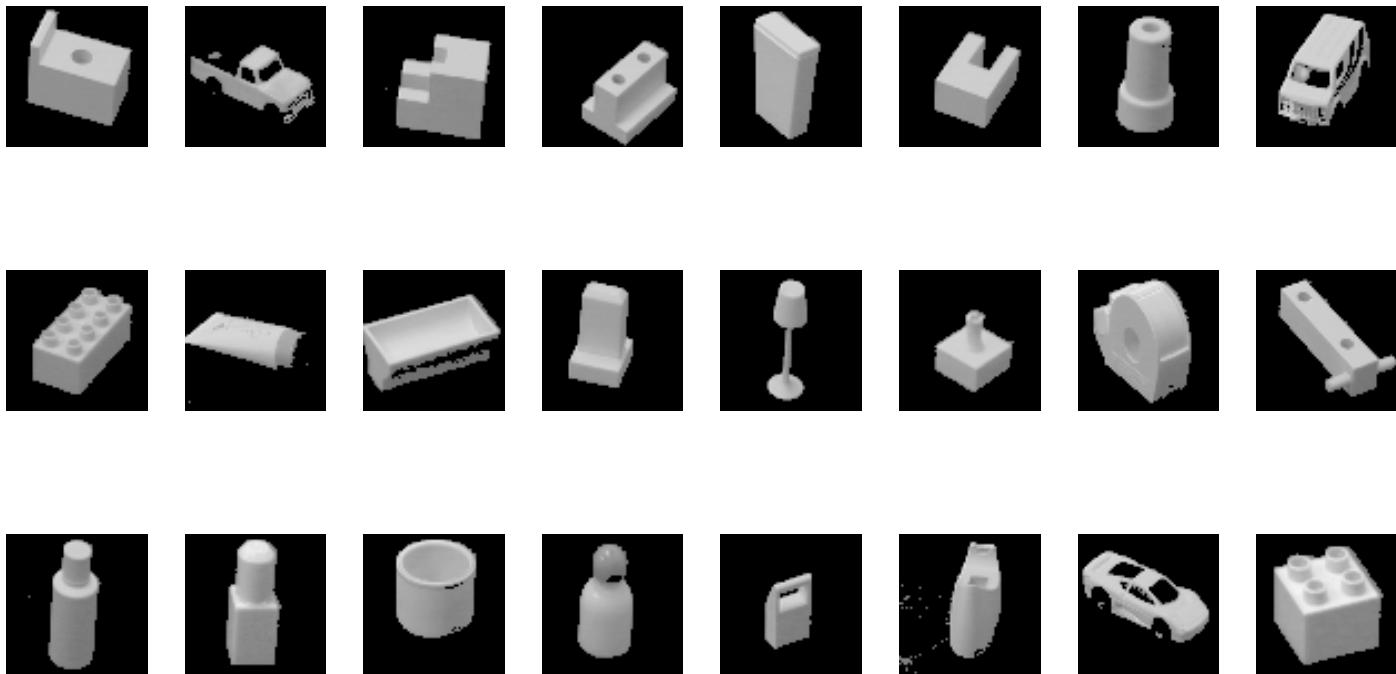
- Make hypotheses if $d < T_1$

RELATIONS:

- Project adjacent regions with $T_1 < d < T_2$ into the ABR eigenspace

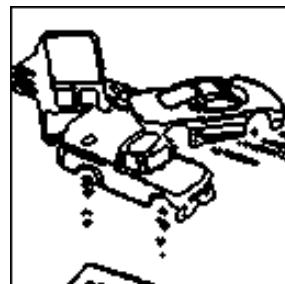
- Make hypotheses if $d' < T_3$

Object Database

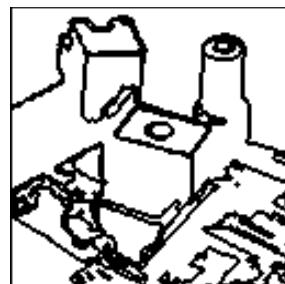
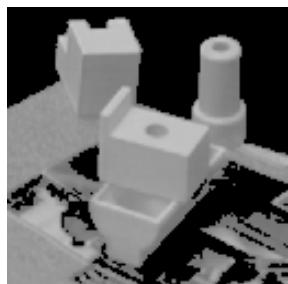


24 objects; 110 ABPs (69 shape groups/3 levels); 130 ABRs

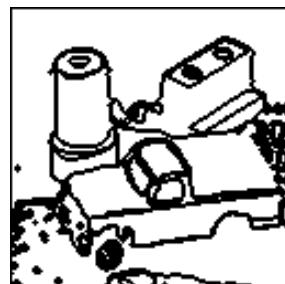
Some Examples



Identity	Score	Pose
Sport Car	0.61	302.77
Truck	0.19	4.29
Ambulance	0.10	11.52



Identity	Score	Pose
Ccube	1.36	41.12
Stamp	0.89	0.64
Holecube	0.89	14.47
Sink	0.68	152.94



Identity	Score	Pose
Stamp	1.91	93.66
Twohole	0.96	181.63
Truck	0.4	93.67

Bag of Words Models

Sivic, Russell, Freeman, Zisserman, ICCV 2005

Fei-Fei and Perona, CVPR 2005

Bosch, Zisserman, Munoz, ECCV 2006

Visual Words

Main idea:

think of image search as a “text retrieval” problem.

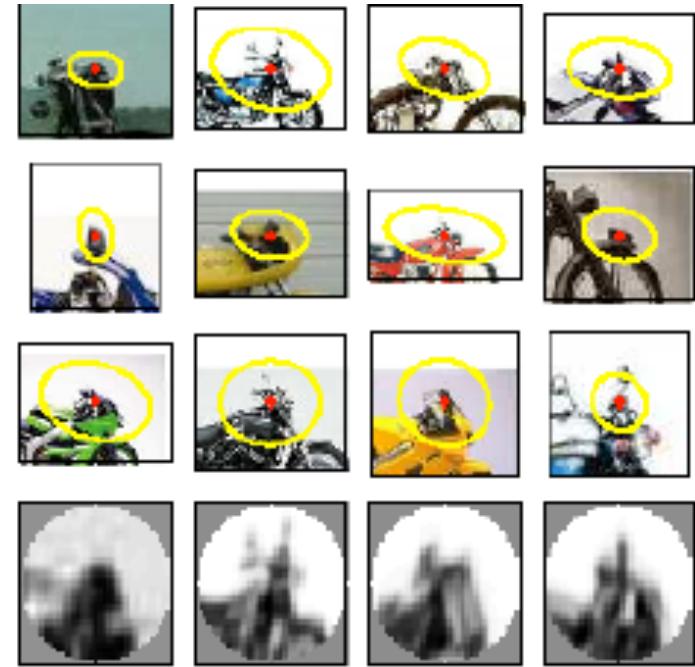
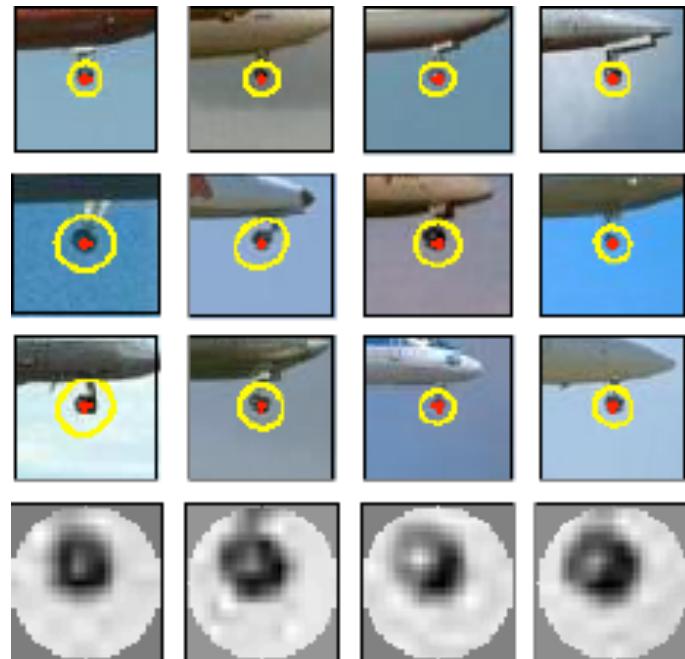
Indexing Local Feature: Inverted File Index

In text documents, we can quickly find all the pages where a word occurs by using an index.

For images, we want to find all images where a feature occurs.

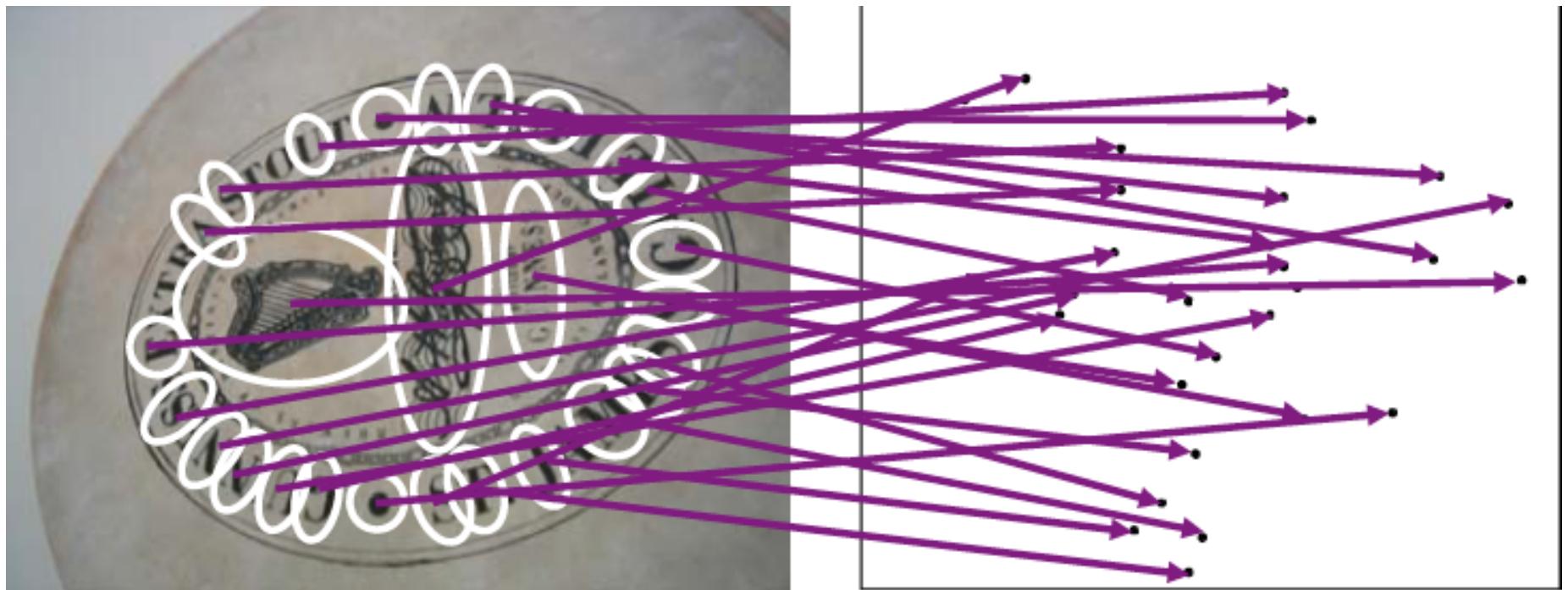
Index	
"Along I-75," From Detroit to Florida; <i>Inside back cover</i>	McGuire, 134
"Drive I-95," From Boston to Florida; <i>Inside back cover</i>	CAA (see AAA)
1929 Green Trail Roadway;	CCC, The; 111,113,115,116,142
101-102,104	Cé d'Art; 147
511 Traffic Information; 63	Caloosahatchee River; 162
AIA (Barrier Is.) - 195 Access; 86	Name; 150
AAA (and CAA); 83	Central Naval Seabore; 173
AAA National Office; 83	Cannon Creek Airport; 130
Abbreviations;	Carry, Read; 105,199
Colored 25 mile Maps; sever	Cape Canaveral; 174
Exit Services; 156	Castillo San Marcos; 169
Tribologue; 85	Cave Diving; 131
Africa; 177	Cayo Costa, Name; 150
Agricultural Inspection Strs.; 125	Celebration; 99
Al-Tan-Thik Museum; 160	Charlotte County; 149
Air Conditioning, First; 112	Charlotte Harbor; 159
Alabama; 124	Chautauqua; 116
Alachua; 132	Chipley; 114
County; 131	Name; 115
Alafia River; 143	Chickees, Name; 115
Alapaha, Name; 120	Circus Museum, Ringling; 147
Alfred B. Maclay Gordon; 106	Citrus; 88,97,130,136,140,180
Alligator Alley; 154-155	CityPlace, W. Palm Beach; 100
Alligator Farm, St.Augustine; 160	City Maps,
Alligator Hole (definition); 157	Ft Lauderdale Expwy; 194-195
Alligator, Bowfin; 150	Jacksonville; 163
Alligators; 100,135,138,147,156	Kissimmee Expwy; 192-193
Anastasia Island; 170	Internal Combustion; 144-149
Arribalzaga; 126-128,146	Orlando Expressways; 192-193
Apalachicola River; 112	Pensacola; 26
Appleton Mus of Art; 136	Tallahassee; 161
Aquifer; 102	Tampa-St. Petersburg; 63
Arabian Nights; 94	St. Augustine; 191
Art Museum, Ringling; 147	Civil War; 100,109,127,138,141
Aruba Beach Cafe; 193	Clearewater Marine Aquarium; 187
Avon River Project; 106	Celler, Baroni; 152
Babcock-Wilcox WMA; 151	Colonial Spanish Quarters; 186
Bahia Mar Marina; 164	Columbia County; 101,128
Baker County; 99	Coquina Building Material; 165
Barefoot Mall; 182	Corkscrew Swamp, Name; 154
Barge Canal; 107	Cowboys; 85
Bell Line Expwy; 80	Cracker, Florida; 88,95,132
Belt Outlet Mall; 89	Cross Trap II; 144
Bernard Castro; 136	Deben Bread; 184
Big "I"; 165	Dade Battlefield; 140
Big Cypress; 155,158	Dade, Maj. Francis; 139-140,161
Big Foot Motel; 105	Dante Beach Hurricane; 184
Bills Swamp Safari; 160	Daniel Boone, Florida Walk; 117
Blackwater River SP; 117	Daytona Beach; 172-173
Blue Angels	De Land; 87
	Diving Lanes; 85
	Dixiel County; 163
	Fair, Roller; 126
	Edison, Thomas; 152
	Eglin AFB; 110-116
	Eight Reels; 176
	Ellenton; 144-145
	Emmanuel Point Wreck; 129
	Emergency Callboxes; 83
	Epiphany; 142,148,167,169
	Escambia Bay; 118
	Bridge (J-10); 119
	County; 120
	Eustis; 153
	Everglades; 93,95,139-140,154-160
	Drawing of; 103,181
	Wildlife MA; 160
	Wender Gardens; 154
	Falling Waters SP; 115
	Fantasy of Flight; 95
	Feyer Dykes SP; 171
	Fires, Forest; 166
	Fires, Prescribed; 146
	Fisherman's Village; 151
	Flagler County; 171
	Flagler, Henry; 97,165,167,171
	Florida Aquarium; 186
	Futura
	12,000 years ago; 107
	Cawm SP; 114
	Mop of all Expressways; 2-3
	Mus of Natural History; 134
	National Cemetery; 141
	Part of Africa; 177
	Platform; 187
	Sheriff's Boys Camp; 126
	Sports Hall of Fame; 130
	Sun 'n Fun Museum; 97
	Supreme Court; 107
	Florida Turnpike (FTP); 178,189
	25 mile Strip Map; 68
	Administration; 129
	Coin System; 150
	Exit Services; 189
	HEFT; 76,161,190
	History; 189
	Names; 159
	Service Plazas; 190
	Spur SR1; 76
	Ticket System; 190
	Toll Plazas; 190
	Feed, Henry; 152

Visual Words



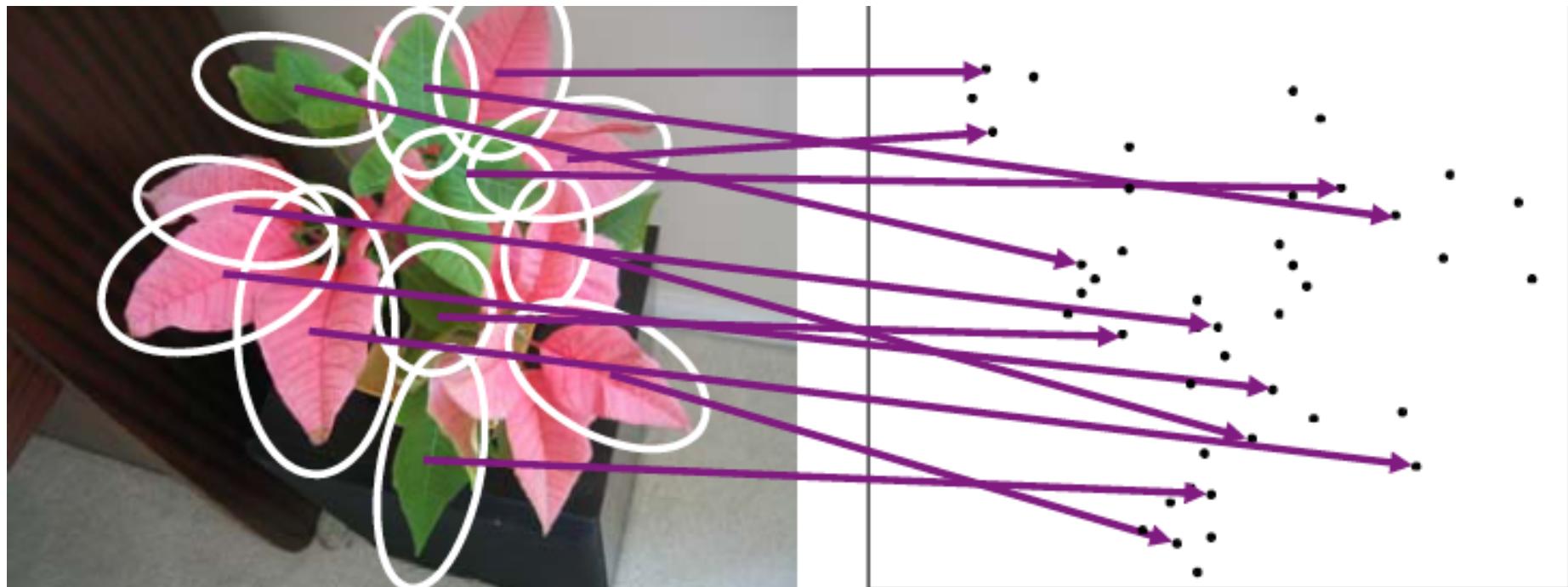
Visual Words: main idea

Extract some local features from a number of images



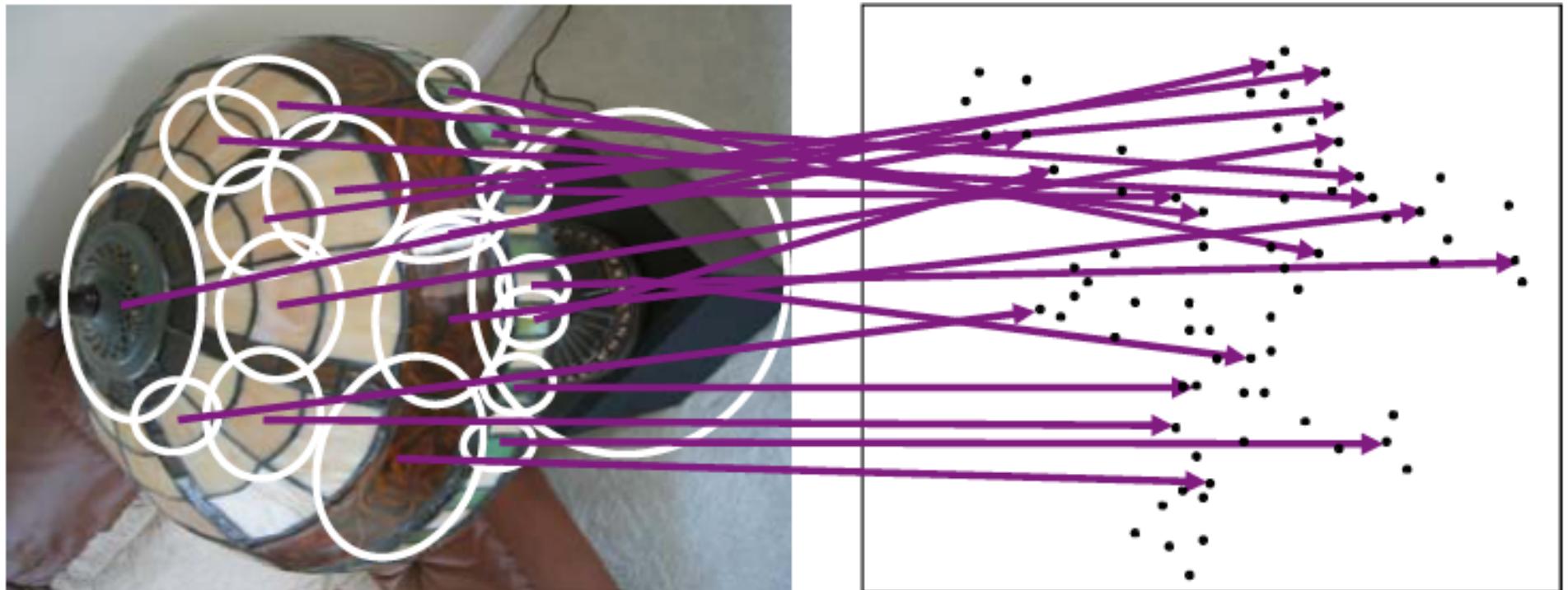
Visual Words: main idea

Extract some local features from a number of images



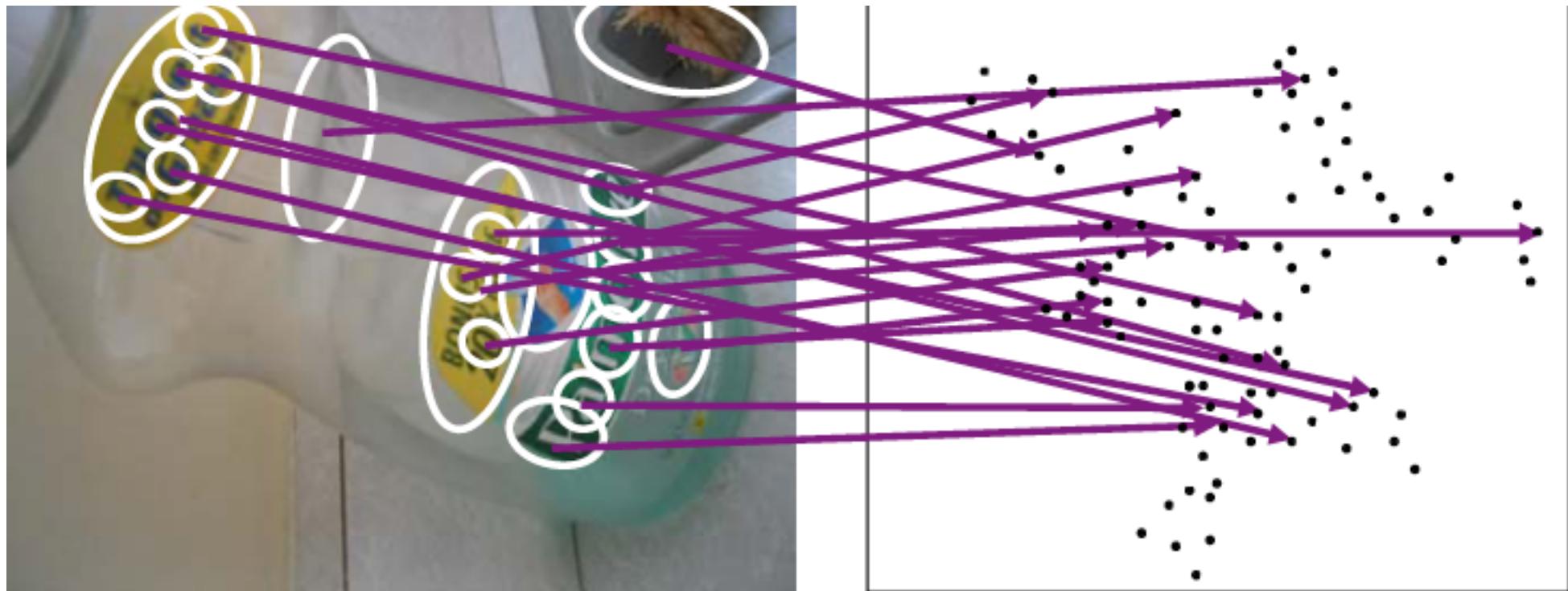
Visual Words: main idea

Extract some local features from a number of images



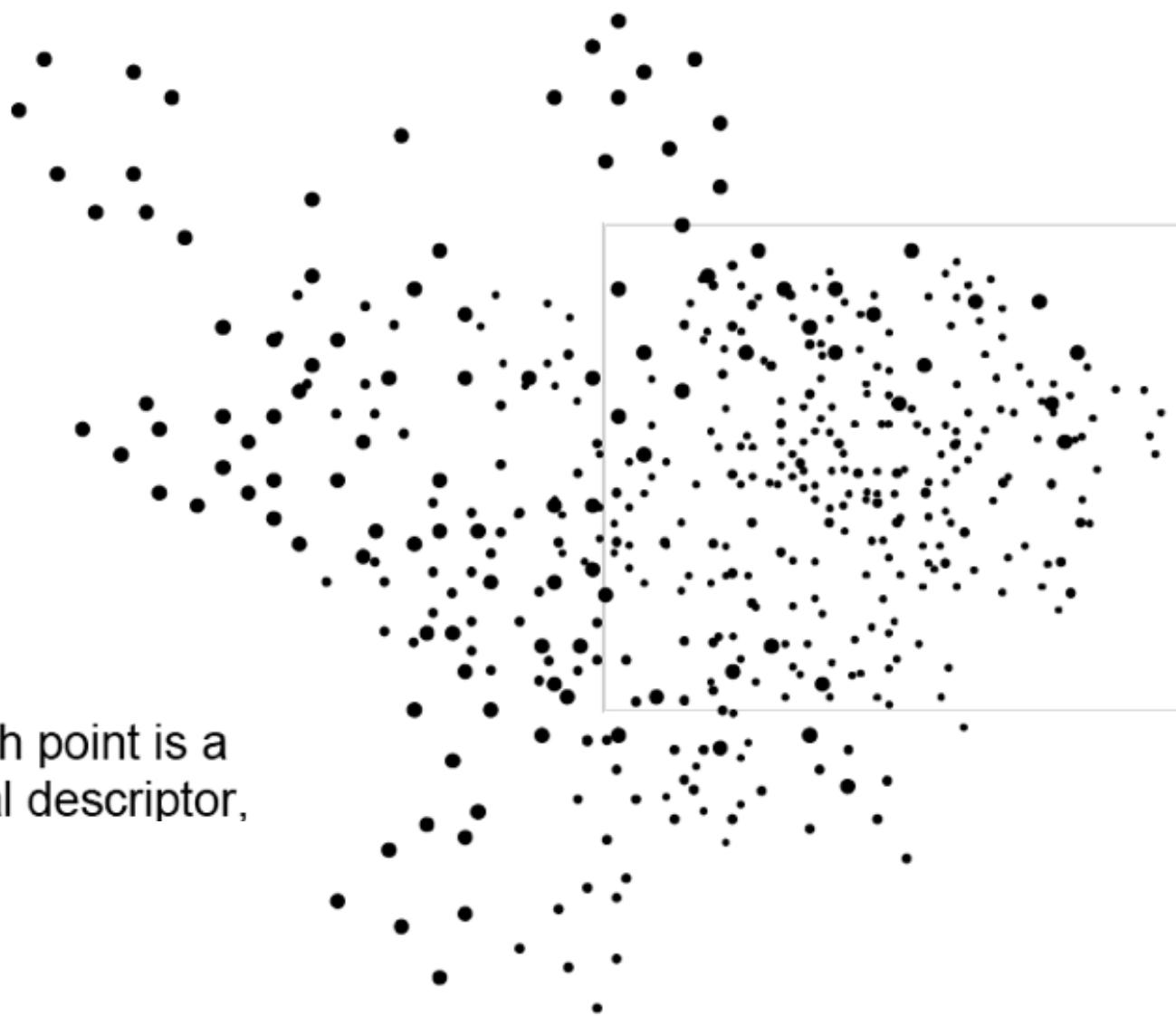
Visual Words: main idea

Extract some local features from a number of images

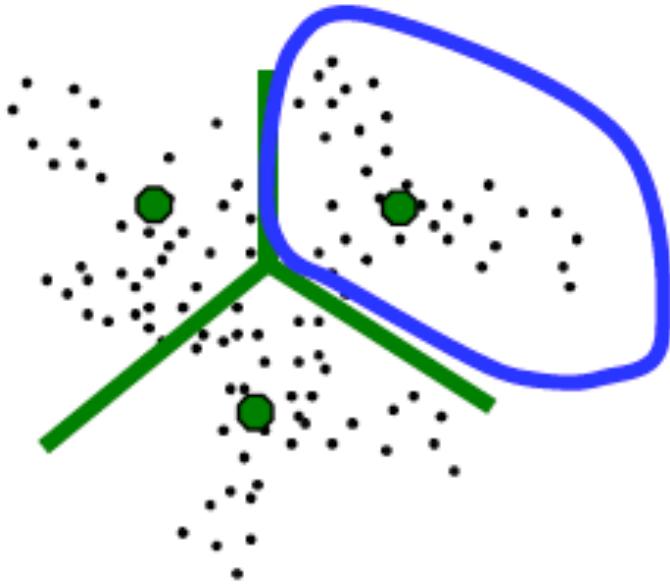


Visual Words: Main Idea

Each point is a local descriptor,



Visual Words



Each group of patches belongs to the same “visual word”.

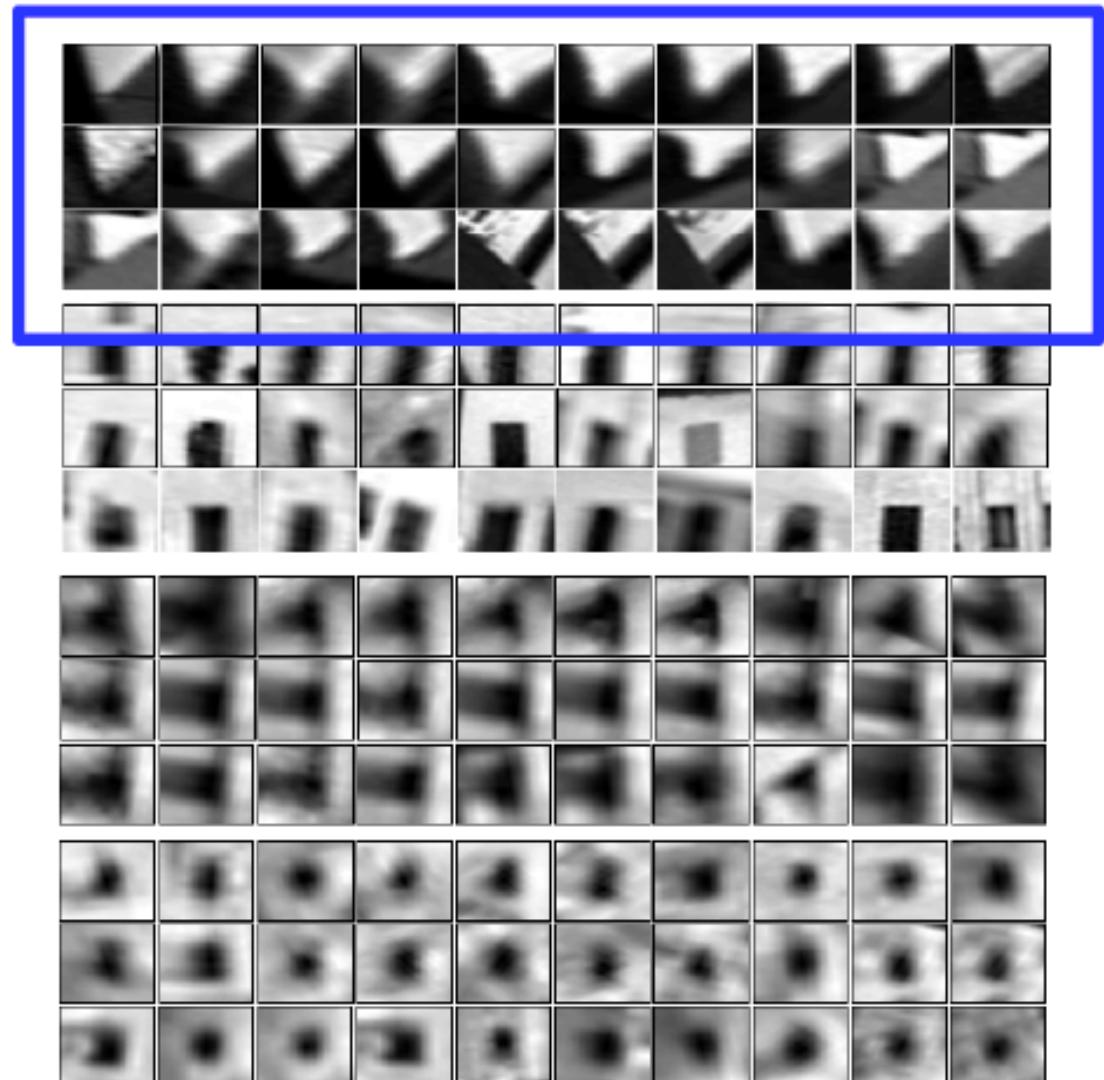
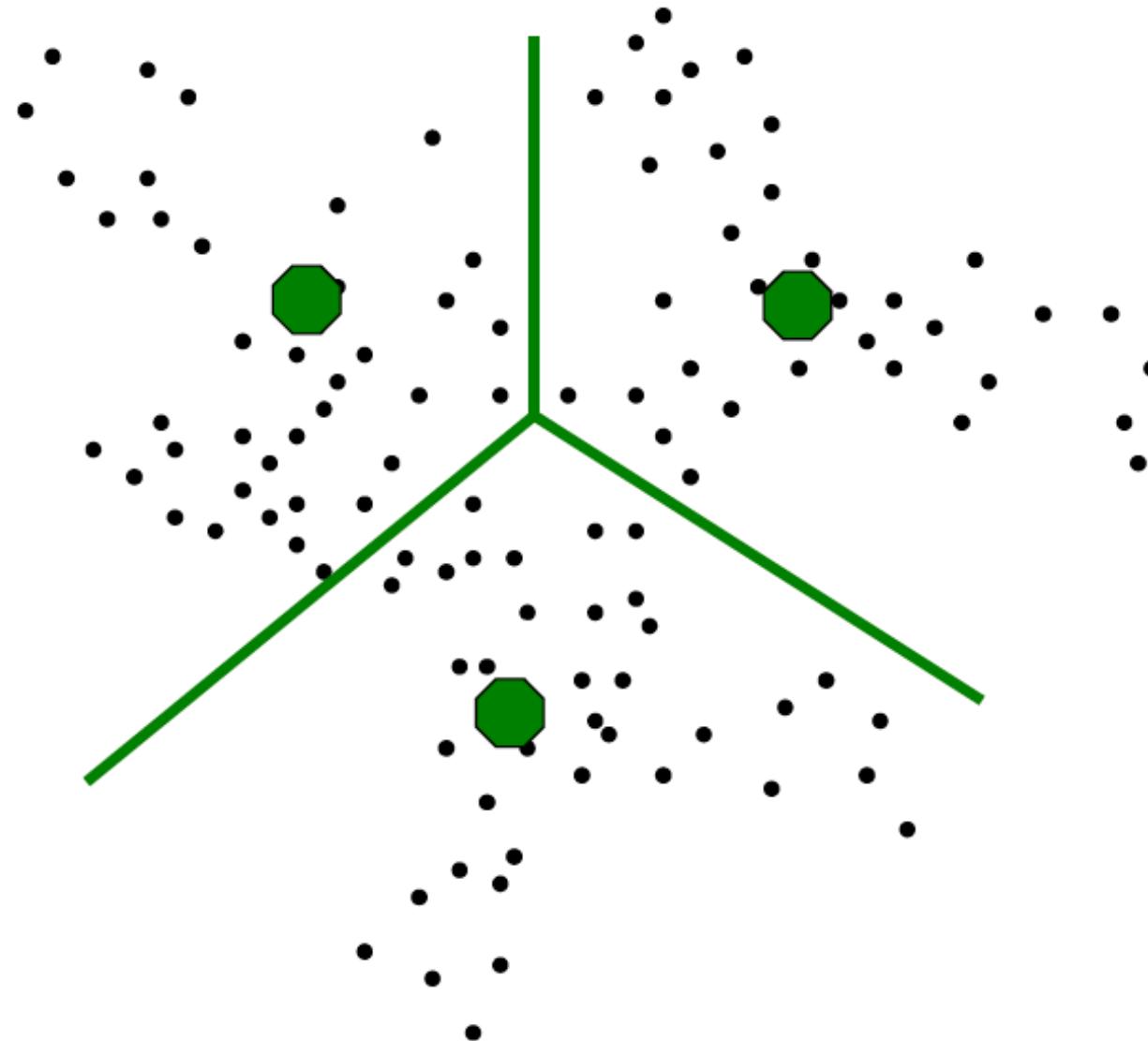


Figure from Sivic & Zisserman, ICCV 2003

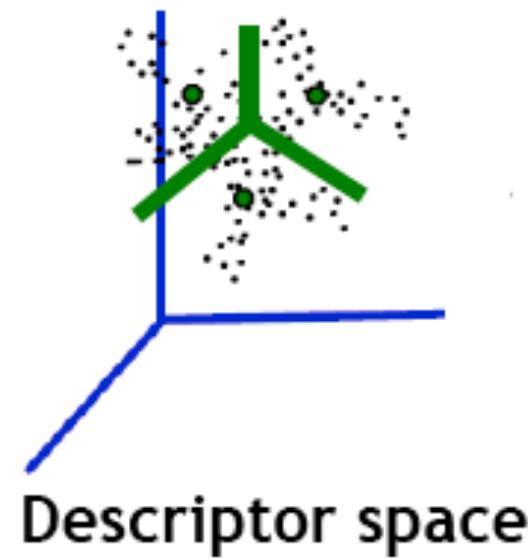
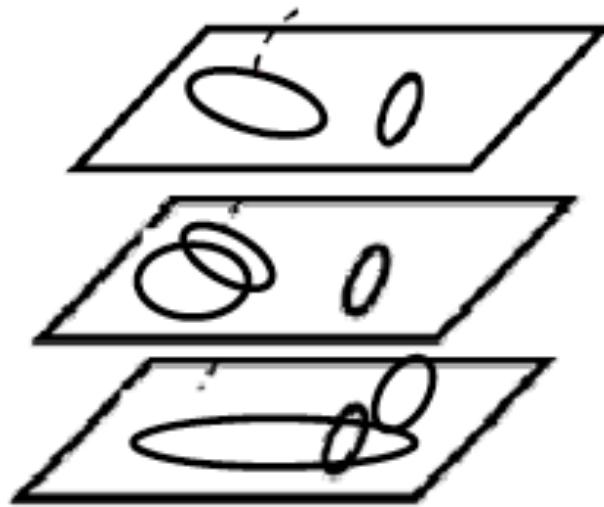
Visual Word Clustering: Vocabulary



Clusters: Prototype Words

Map high dimensional descriptors to tokens/words by quantizing the feature space:

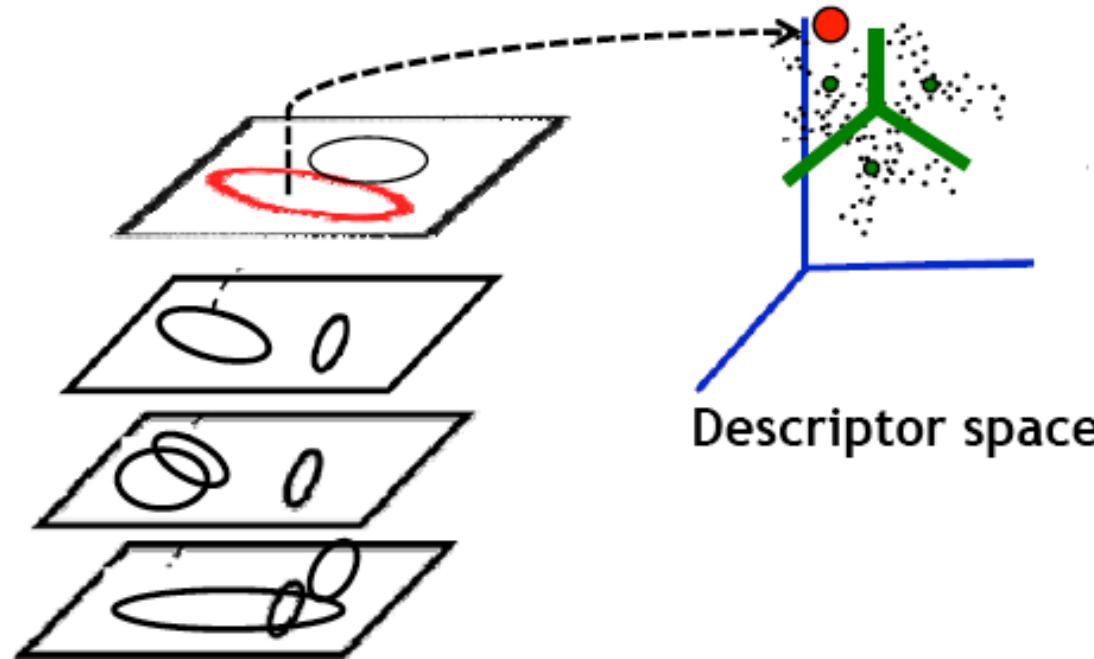
Quantize by clustering: each cluster center is a word prototype



Clusters: Prototype Words

Map high dimensional descriptors to tokens/words by quantizing the feature space:

Determine which word to assign to each new image by finding the closest cluster center



Inverted File Index

Database images are loaded into the index mapping words to image numbers

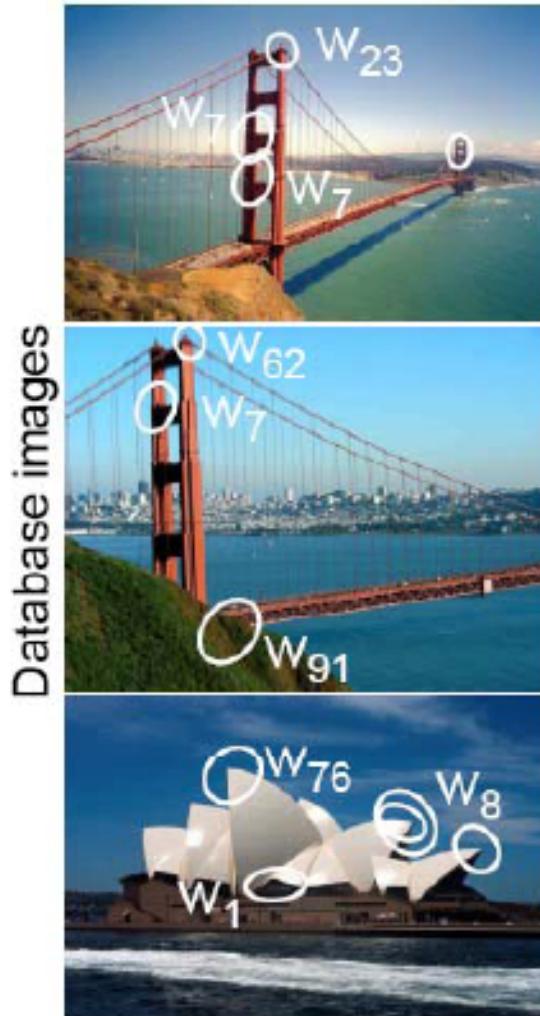


Image #1

Image #2

Image #3

Word #	Image #
1	3
2	
...	
7	1, 2
8	3
...	
9	
10	
...	
91	2

...

Inverted File Index

New query image is mapped to indices of database images that share a word.



New query image

Word #	Image #
1	3
2	
...	
7	1, 2
8	3
9	
10	
...	
91	2

Visual Documents

If a local region is a “visual word”, how can we summarize an image - i.e. a “visual document”?

Visual Documents

Of all the sensory impressions proceeding to the brain, the visual experiences are the dominant ones. Our perception of the world around us is based essentially on the messages that reach our brain from our eyes. For a long time it was believed that the retinal image was processed by visual centers in the cerebral cortex, as a movie screen displays a sequence of image frames. In 1960, two American scientists, David Hubel and Torsten Wiesel, discovered that the visual system is more complex than previously thought. Following the path of the optic nerve through the brain to the various cortical areas, they found that the cortex, Hubel and Wiesel have demonstrated that the message about the image falling on the retina undergoes a column-wise analysis in a system of nerve cells stored in columns. In this system each column has its specific function and is responsible for a specific detail in the pattern of the retinal image.

**sensory, brain,
visual, perception,
retinal, cerebral cortex,
eye, cell, optical
nerve, image
Hubel, Wiesel**

China is forecasting a trade surplus of \$90bn (£51bn) to \$100bn this year, a threefold increase on 2004's \$32bn. The Commerce Ministry said the surplus would be created by a predicted 30% increase in exports to \$750bn, compared with \$660bn. That figure would annoy the US, which wants China's central bank to deliberately devalue the yuan to ease its trade surplus. The Chinese government also needs to encourage foreign demand so that it can diversify its economy. China has been allowed to let the yuan against the dollar rise slowly and permitted it to trade within a narrow band, but the US wants the yuan to be allowed to trade freely. However, Beijing has made it clear that it will take its time and tread carefully before allowing the yuan to rise further in value.

**China, trade,
surplus, commerce,
exports, imports, US,
yuan, bank, domestic,
foreign, increase,
trade, value**

ICCV 2005 short course, L. Fei-Fei

Object

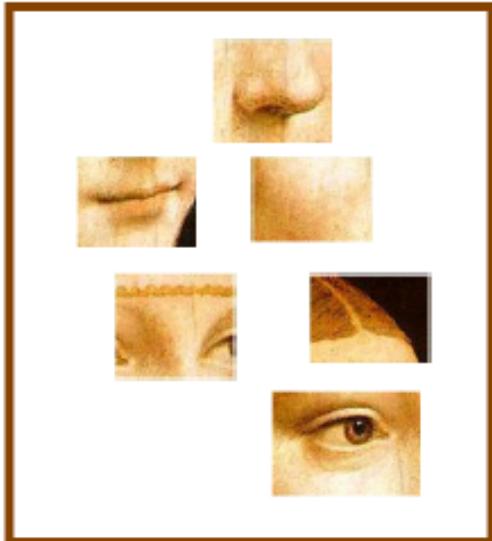
Bag of 'words'



Definition of “BoW”

Independent features

face



bike

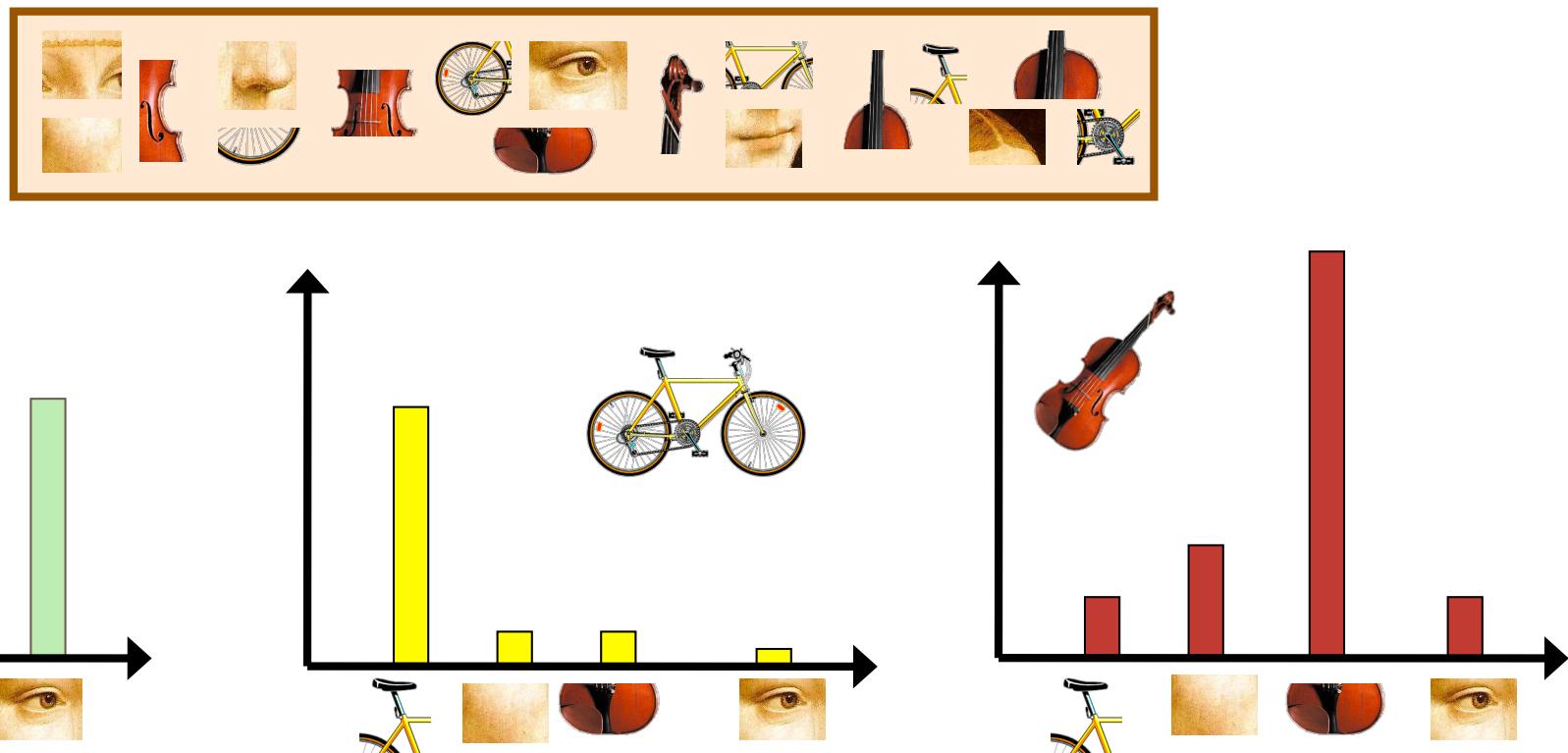


violin



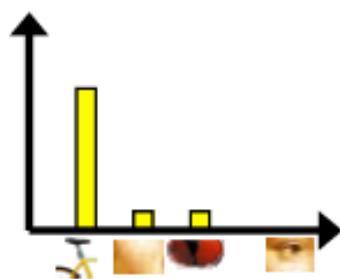
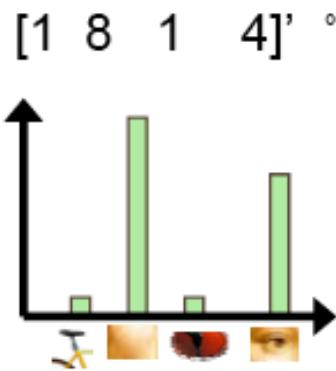
Definition of “BoW”

Independent features
histogram representation



Comparing BoW

Rank frames by normalized scalar product between their (possibly weighted) occurrence counts -- nearest neighbor search for similar images.



$$\vec{d}_j \quad \vec{q}$$

$$\text{sim}(d_j, q) = \frac{d.q}{|d||q|} = \frac{\sum_{i=1}^t w_{ji} w_{qi}}{\sqrt{\sum_{i=1}^t w_{ji}^2} \sqrt{\sum_{i=1}^t w_{qi}^2}}$$

tf-idf weighting

Term Frequency - Inverse Document Frequency:

Describe frame by frequency of each word in it

Downweight words that appear often in the database

Standard weighting in text retrieval

$$t_i = \frac{n_{id}}{n_d} \log \frac{N}{n_i}$$

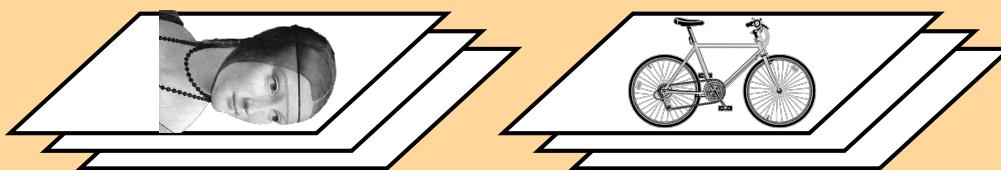
Number of occurrences of word i in doc d .

Number words in doc d .

Total number of docs in database

Number of docs where word i occurs in, in the whole database

learning



feature detection
& representation

codewords dictionary

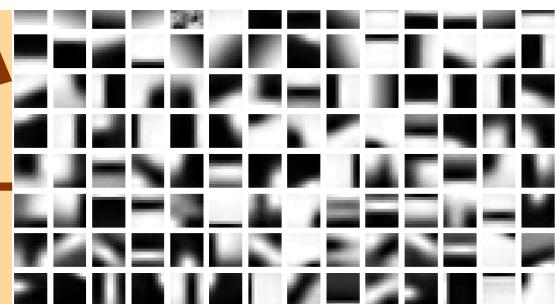
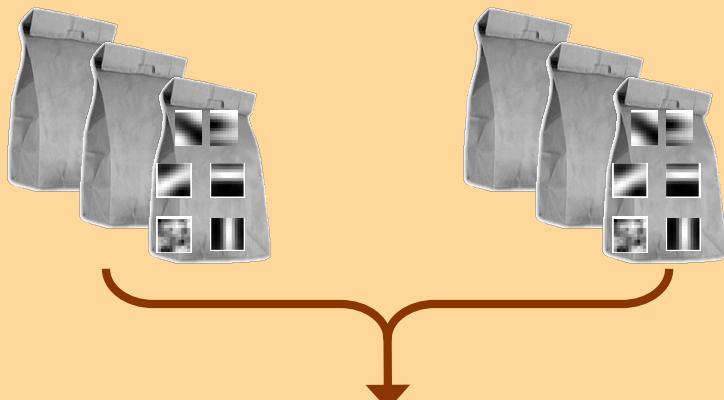
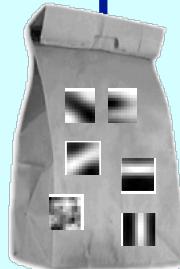


image representation



**category models
(and/or) classifiers**

recognition



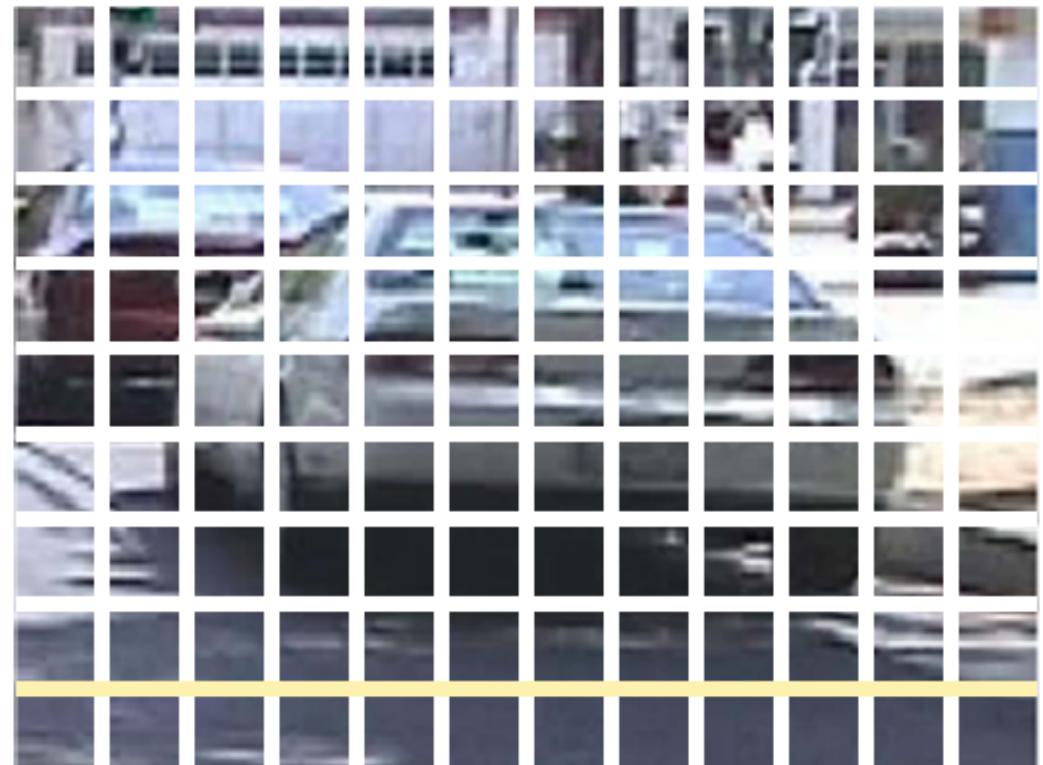
**category
decision**

Feature Detection

Regular grid

Vogel & Schiele, '03

Fei-Fei & Perona '05



Feature Detection

Regular grid

Vogel & Schiele, '03

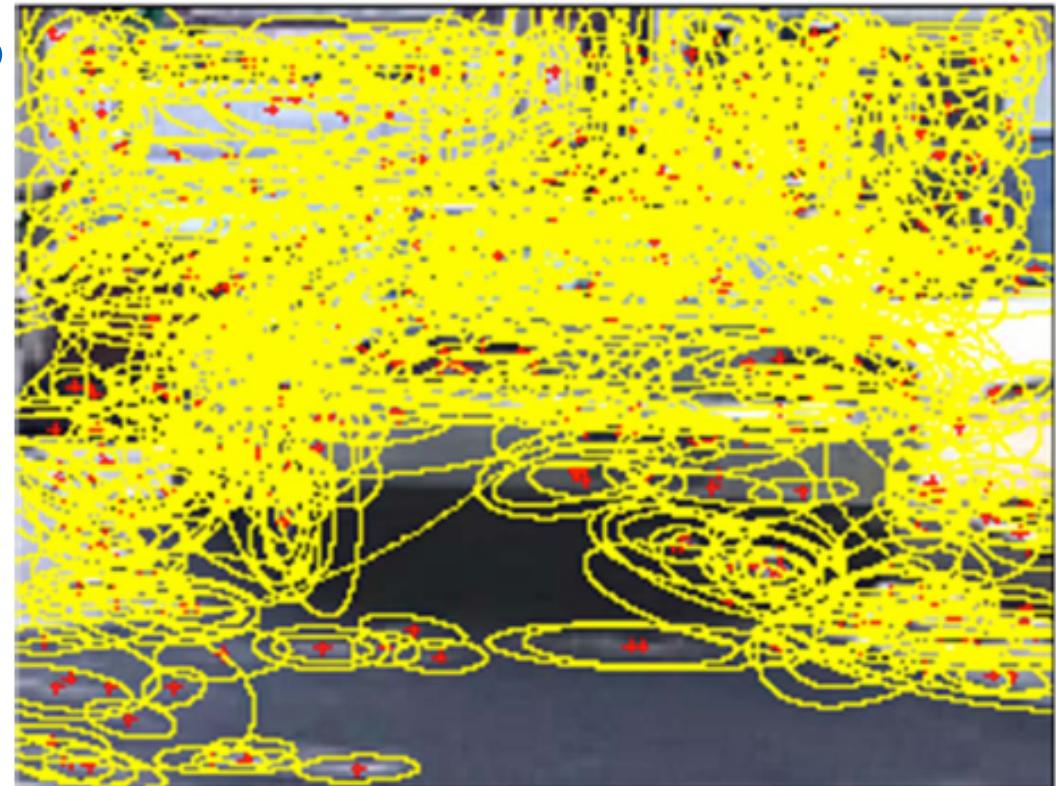
Fei-Fei & Perona '05

Interest Points

Csurka et al, '04

Fei-Fei & Perona '05

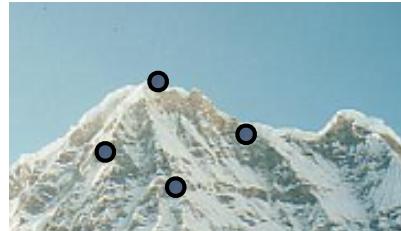
Sivic et al '05



Matching with Features

Problem 1:

Detect the same points in both images

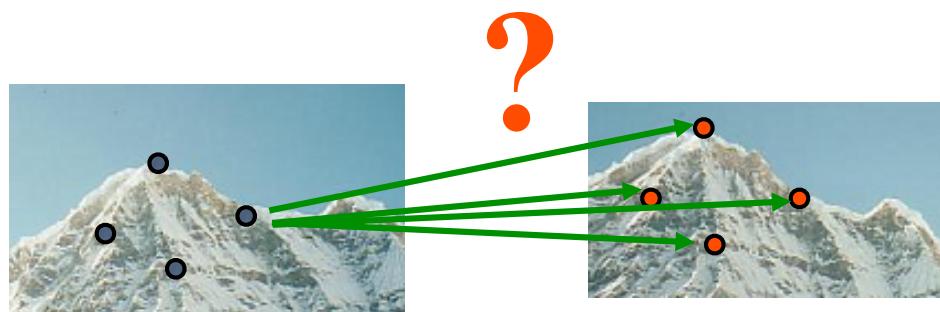


Feature detection must be repeatable

Matching with Features

Problem 2:

Match features correctly



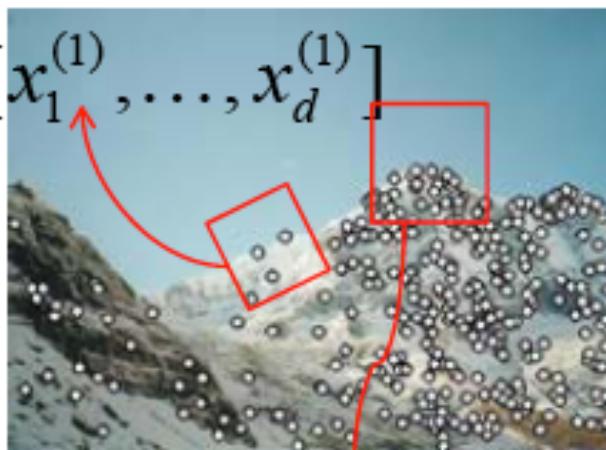
We need a good description of the features

Using Local Features

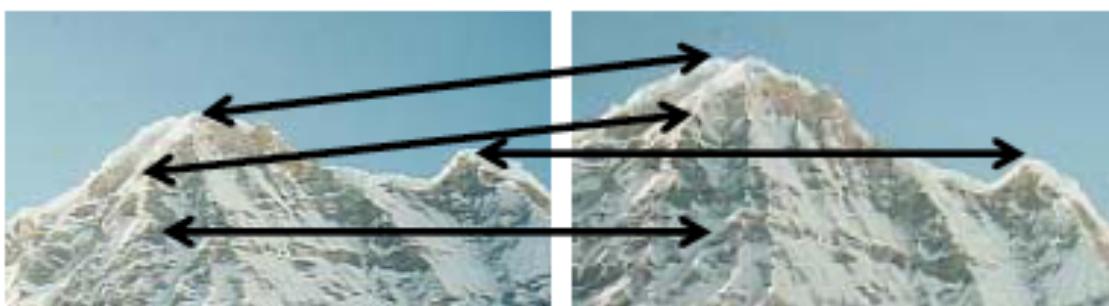
1. Detection: Identify interesting points
2. Description: Extract vectors describing the local features around the interesting points
3. Matching: Determine correspondence between descriptors in two views



$$\mathbf{x}_1 = [x_1^{(1)}, \dots, x_d^{(1)}]$$



$$\mathbf{x}_2 = [x_1^{(2)}, \dots, x_d^{(2)}]$$



Local Features: Desired Properties

Repeatability

The same feature can be found in several images, despite geometric and photometric transformations

Saliency

Each feature has a distinctive description

Compactness and Efficiency

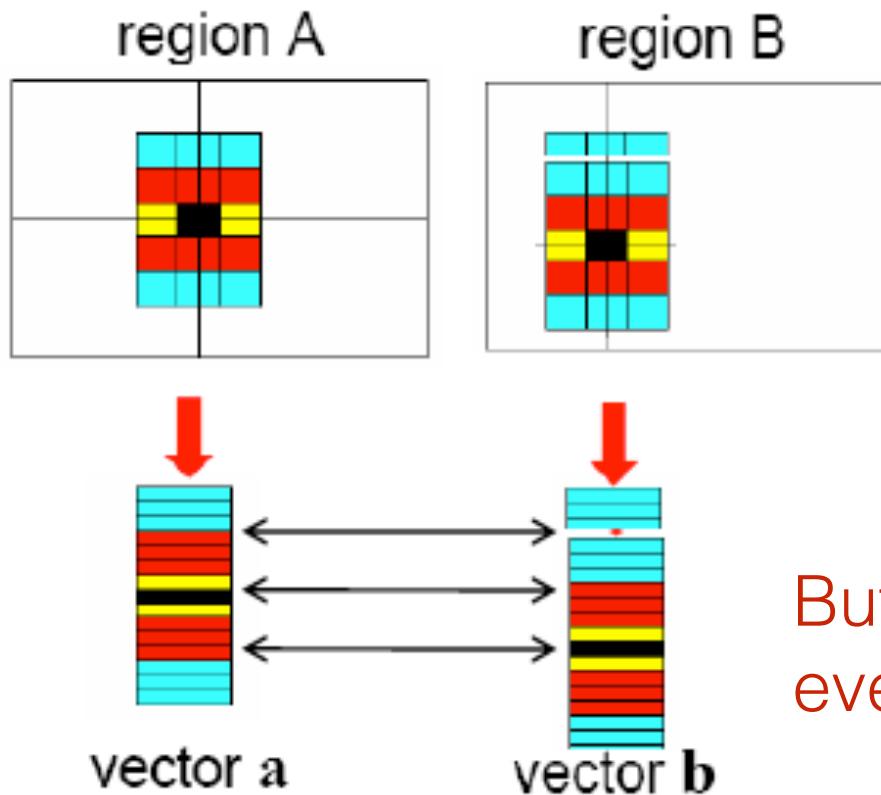
Lot less features than pixels

Locality

A feature occupies a relatively small area

Robust to occlusion and clutter

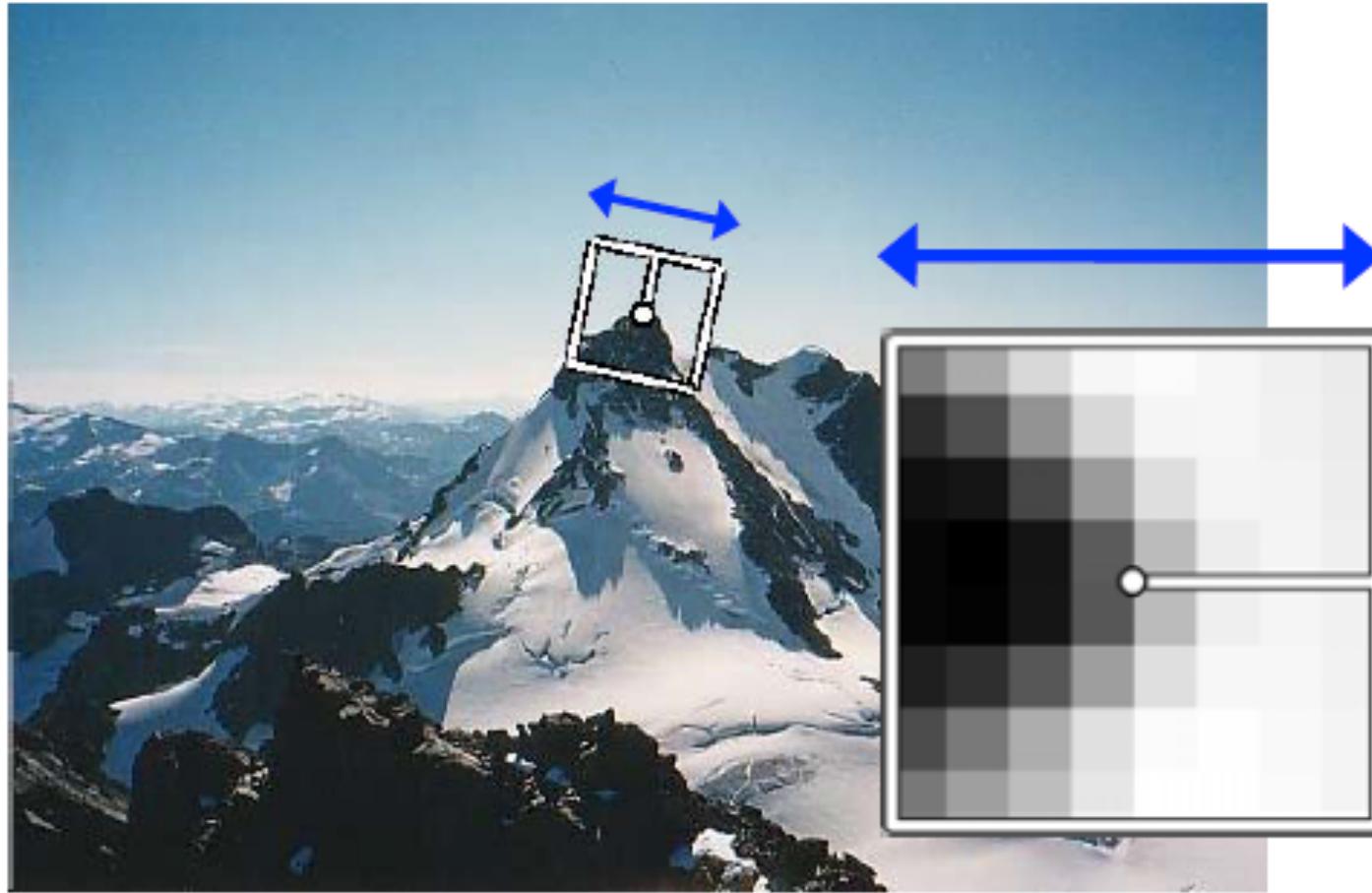
Raw Patches as Local Descriptors



The simplest way to describe the neighborhood around an interest point is to write down the list of intensities to form a feature vector.

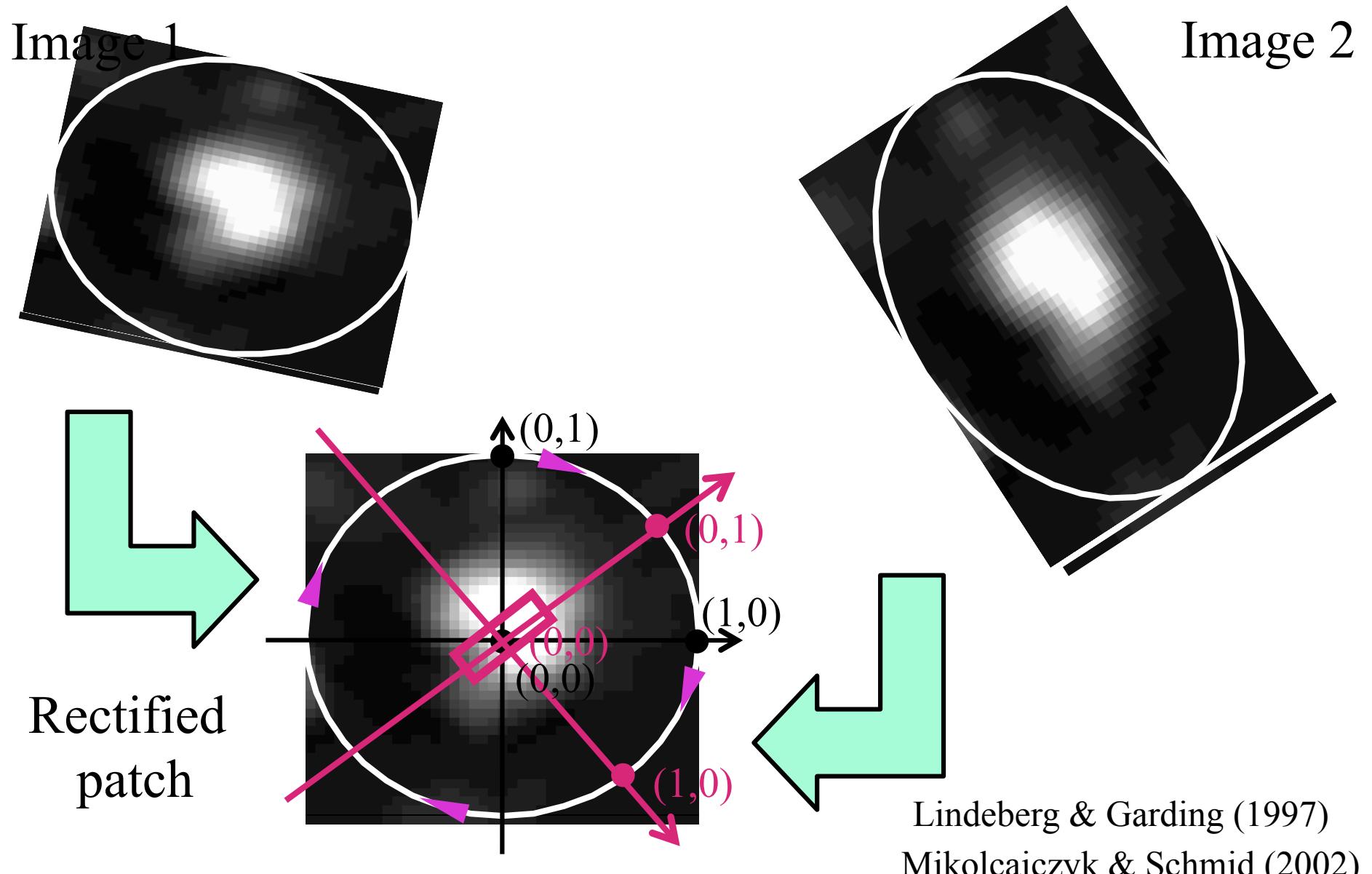
But this is very sensitive to even small shifts and rotations

Rotation Invariance

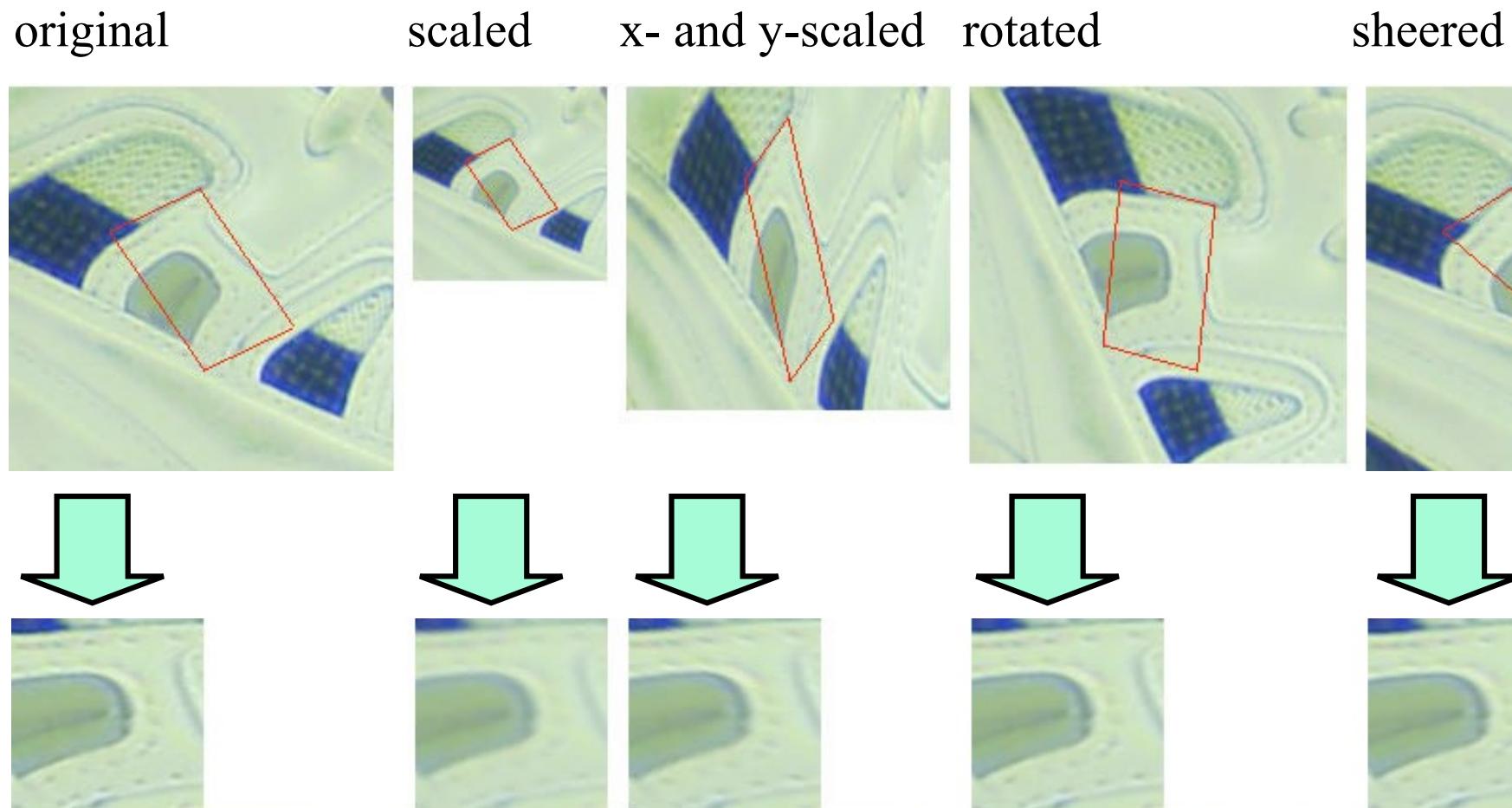


Rotate patch according to its dominant gradient orientation:
“canonical” representation.

Affine adaptation/Rectification process



Rectification Example



SIFT - Scale Invariant Feature Transform

D.Lowe. "Distinctive Image Features from Scale-Invariant Keypoints".
IJCV 2004

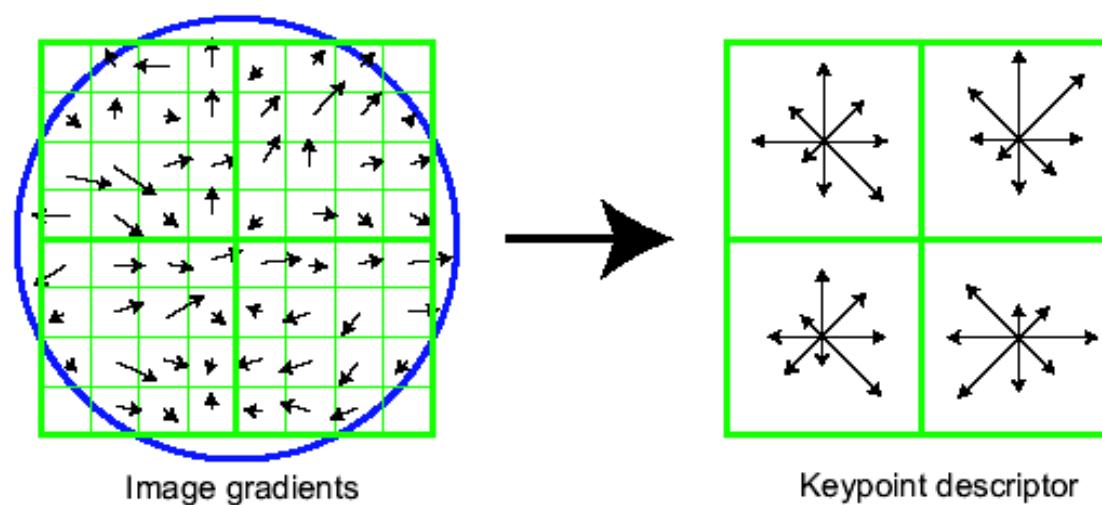
Descriptor overview:

Determine scale (by maximizing DoG in scale and in space),
local orientation as the dominant gradient direction.

Use this scale and orientation to make all further computations invariant to scale and rotation.

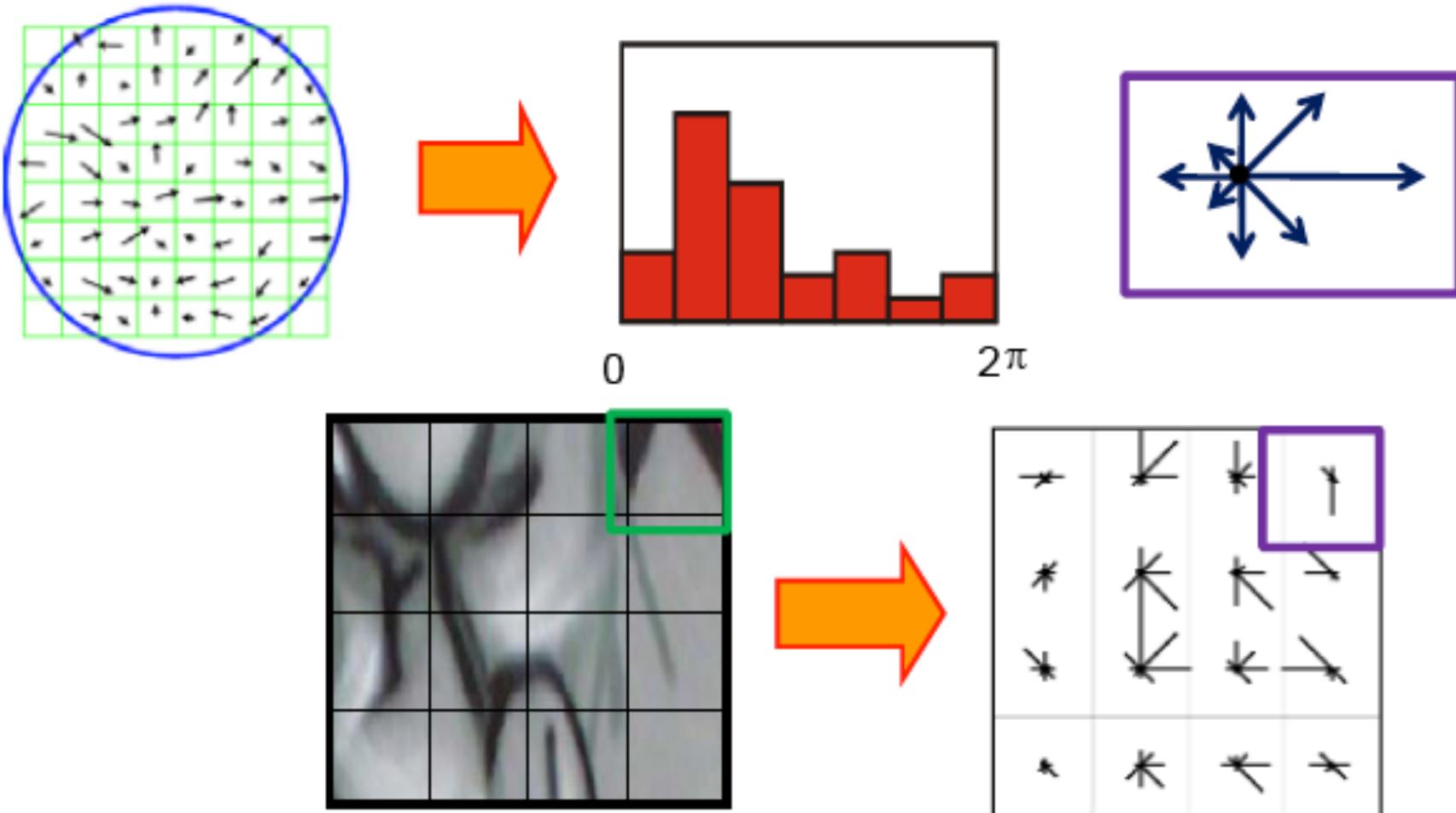
Compute gradient orientation histograms of several small windows (128 values for each point)

Normalize the descriptor to make it invariant to intensity change



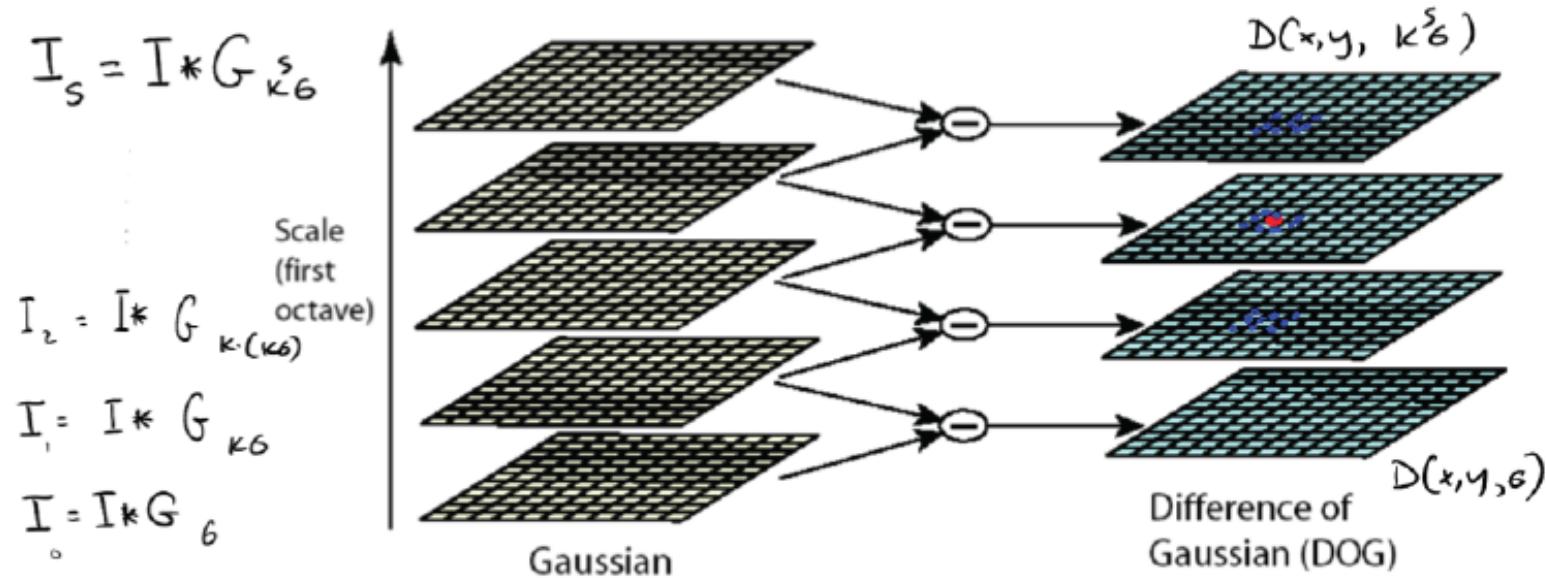
SIFT DESCRIPTOR (Lowe 2004)

Use histograms to bin pixels within sub-patches according to their orientation.



SIFT Descriptor

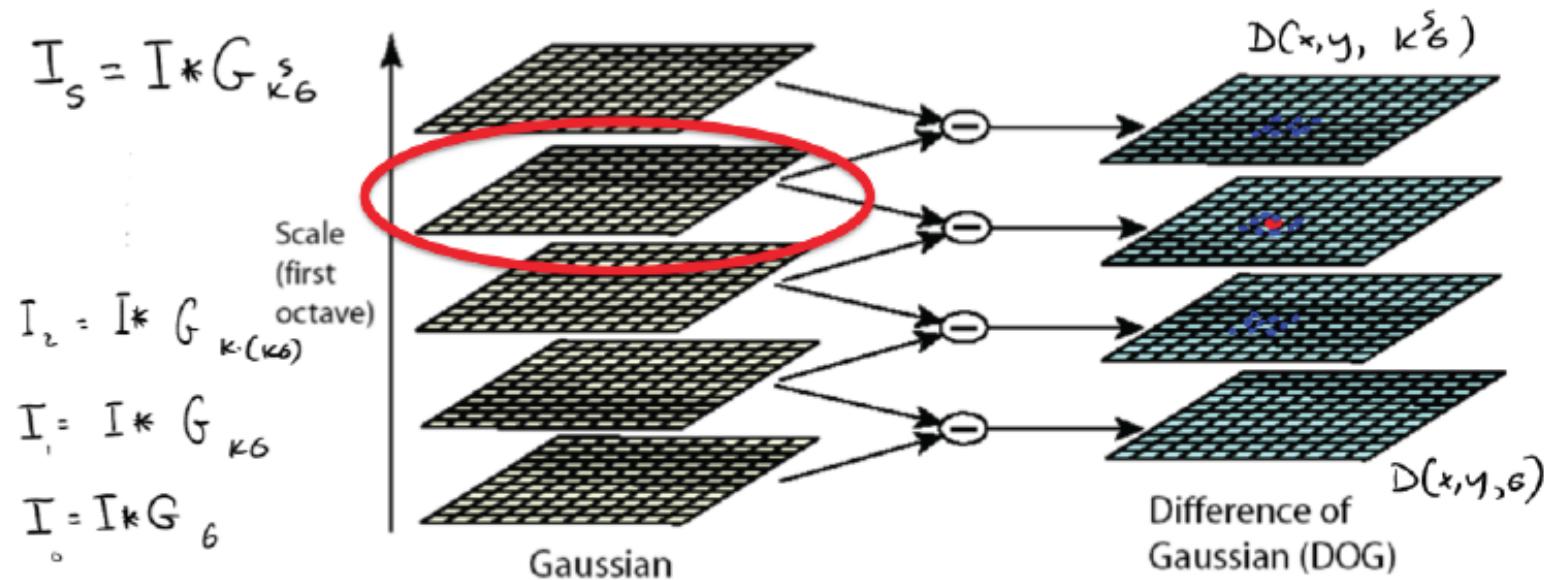
- ① Our scale invariant interest point detector gives scale ρ for each keypoint



[Adopted from: F. Flores-Mangas]

SIFT Descriptor

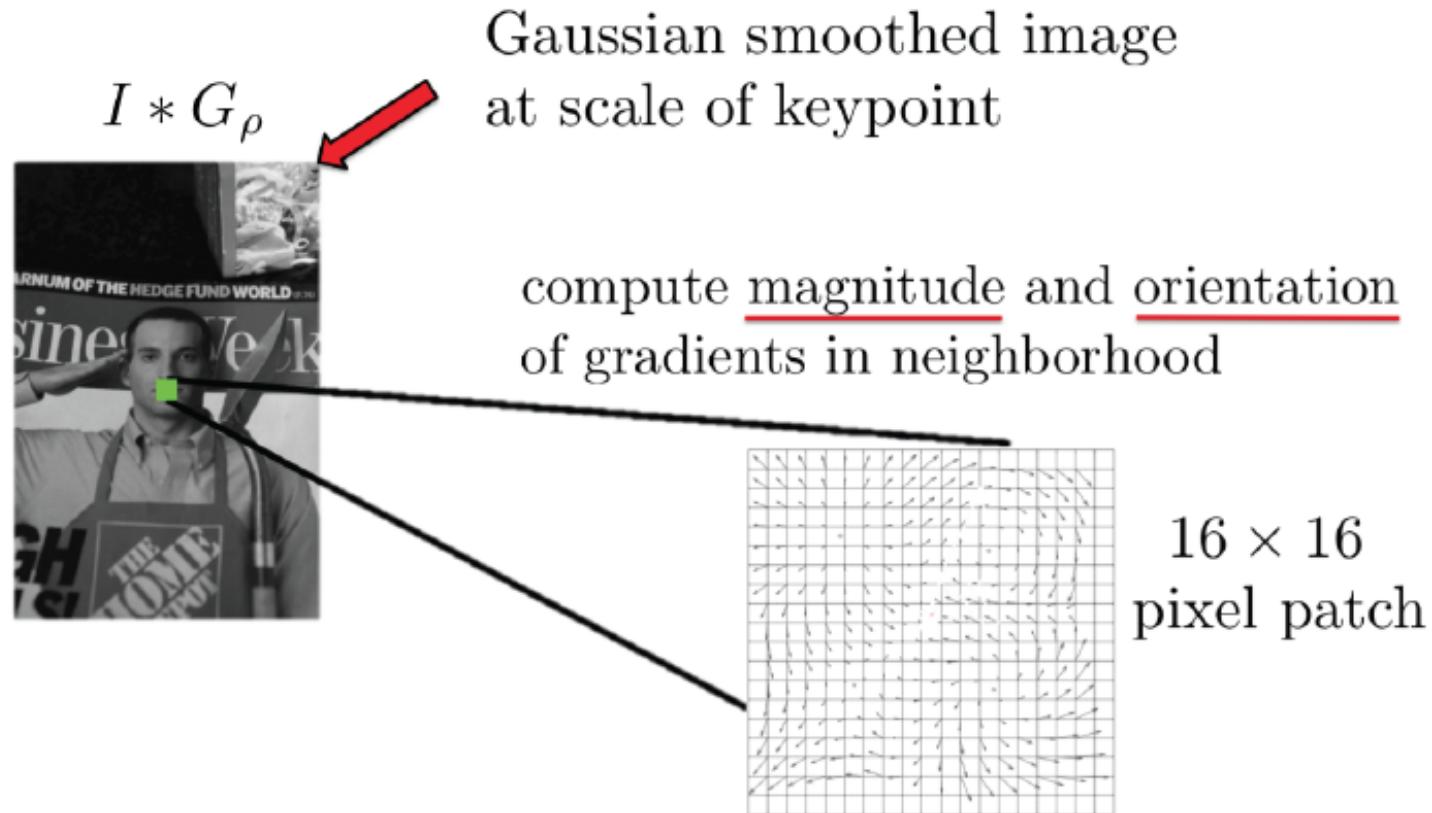
- ② For each keypoint, we take the Gaussian-blurred image at corresponding scale ρ



[Adopted from: F. Flores-Mangas]

SIFT Descriptor

- ③ Compute the gradient magnitude and orientation in neighborhood of each keypoint



[Adopted from: F. Flores-Mangas]

SIFT Descriptor

- ③ Compute the gradient magnitude and orientation in neighborhood of each keypoint

magnitude of gradient:

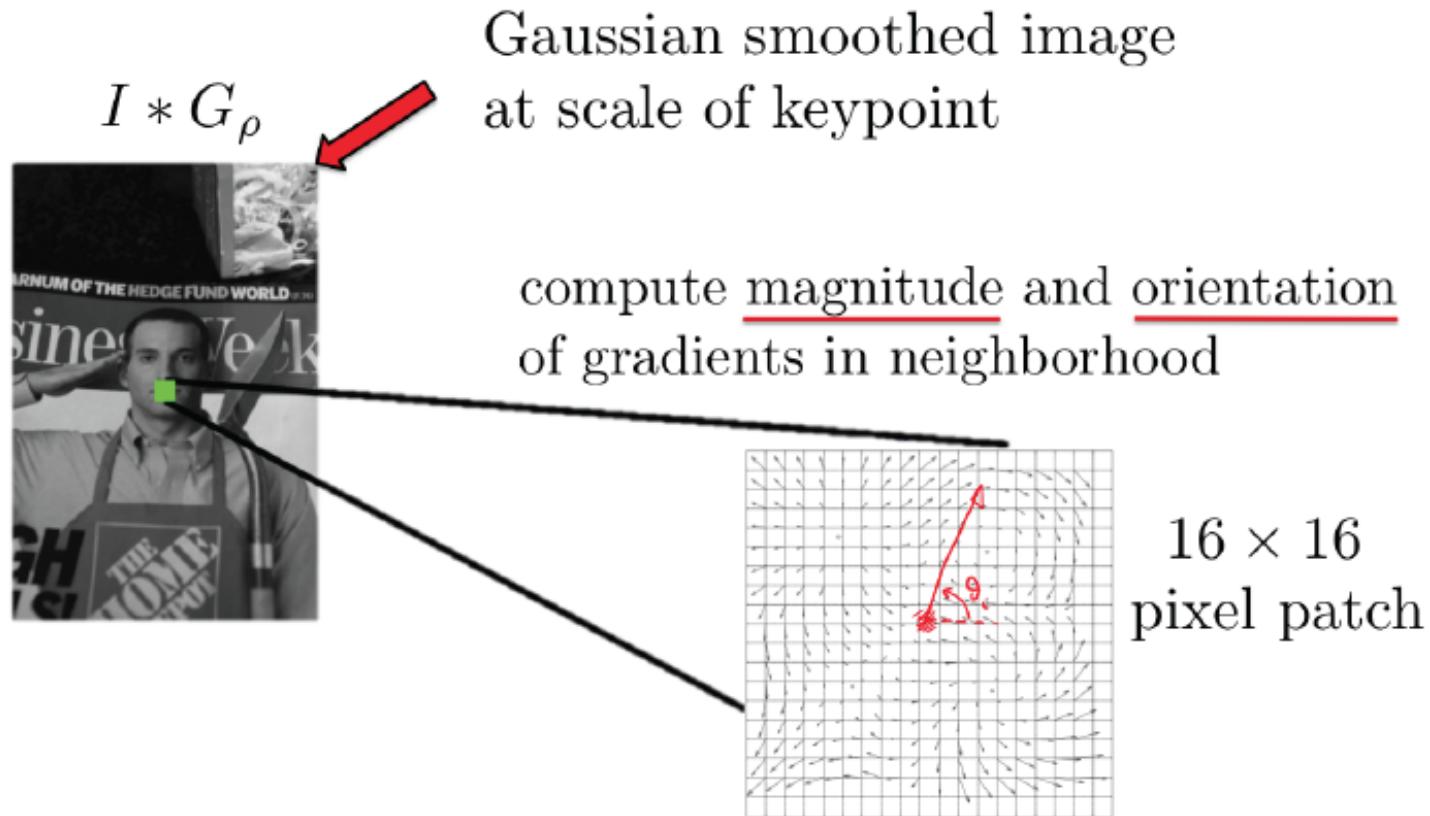
$$|\nabla I(x, y)| = \sqrt{\left(\frac{\partial(I(x, y) * G_\rho)}{\partial x}\right)^2 + \left(\frac{\partial(I(x, y) * G_\rho)}{\partial y}\right)^2}$$

gradient orientation:

$$\theta(x, y) = \arctan\left(\frac{\partial I * G_\rho}{\partial y} / \frac{\partial I * G_\rho}{\partial x}\right)$$

SIFT Descriptor

- ④ Compute dominant orientation of each keypoint. How?

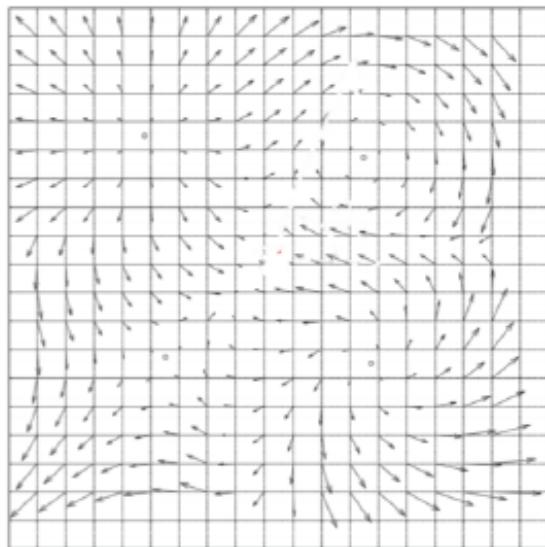


[Adopted from: F. Flores-Mangas]

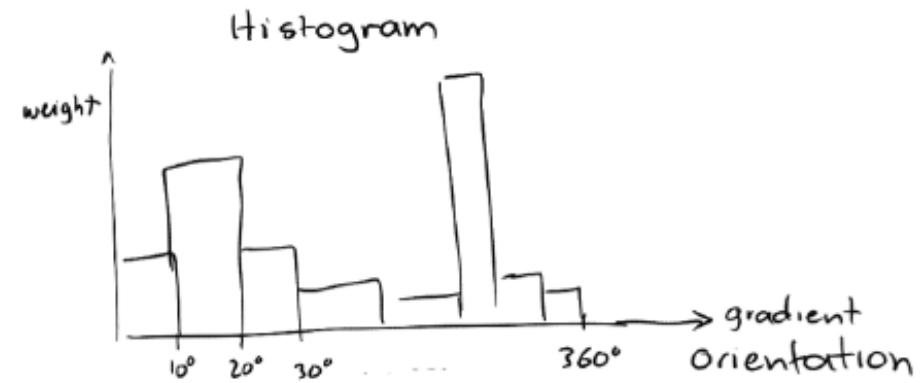
SIFT Descriptor

- Compute a histogram of gradient orientations, each bin covers 10°

16×16



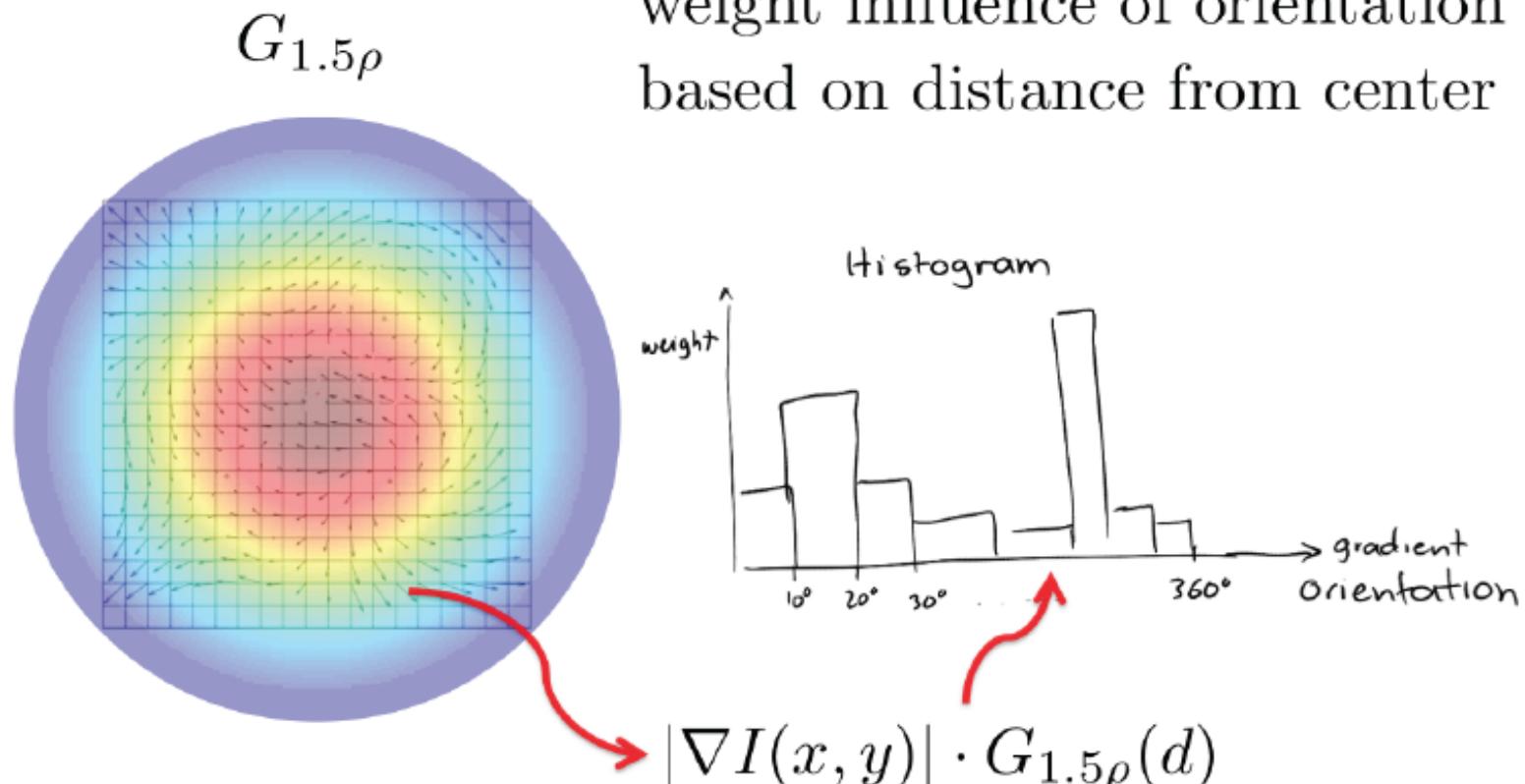
compute histograms of orientations
by orientation increments of 10°



[Adopted from: F. Flores-Mangas]

SIFT Descriptor: Dominant Orientation

- Compute a histogram of gradient orientations, each bin covers 10°
- Orientations closer to the keypoint center should contribute more

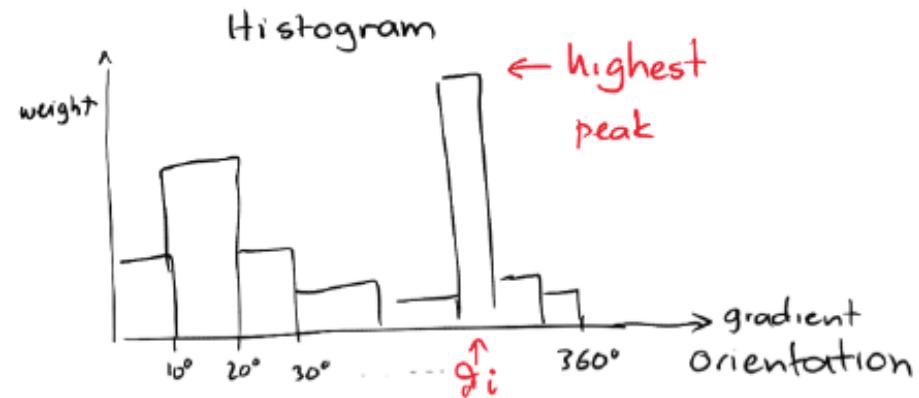
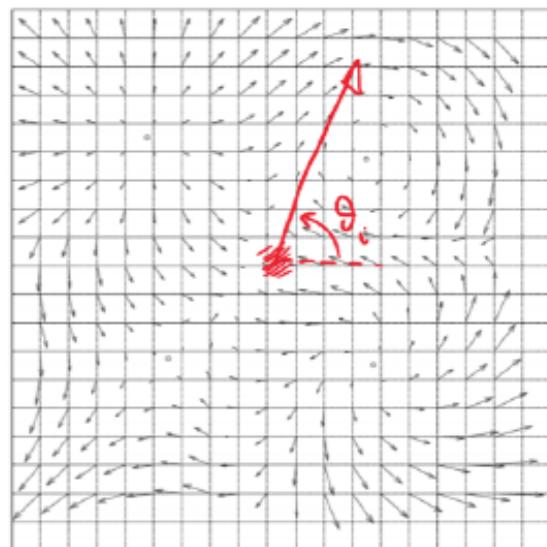


[Adopted from: F. Flores-Mangas]

SIFT Descriptor: Dominant Orientation

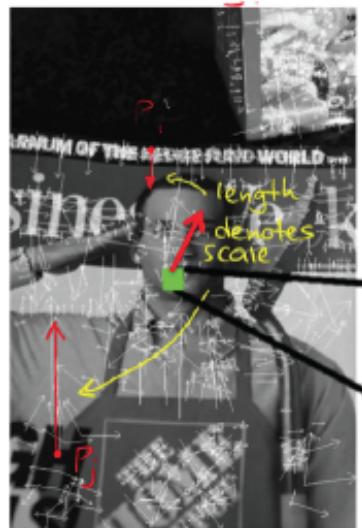
- Compute a histogram of gradient orientations, each bin covers 10°
- Orientations closer to the keypoint center should contribute more
- Orientation giving the peak in the histogram is the keypoint's orientation

16×16

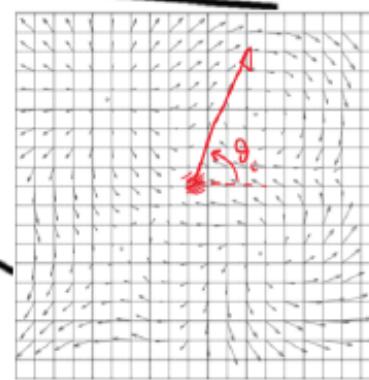


SIFT Descriptor

④ Compute dominant orientation



compute magnitude and orientation
of gradients in neighborhood

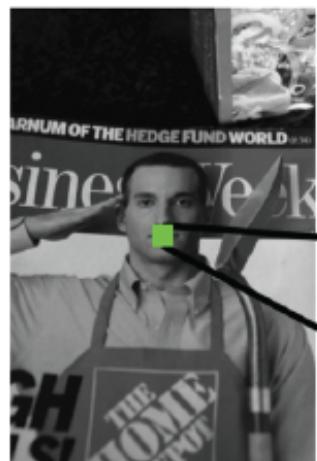


16×16
pixel patch

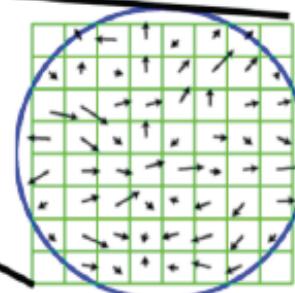
[Adopted from: F. Flores-Mangas]

SIFT Descriptor

- ⑤ Compute a 128 dimensional descriptor: 4×4 grid, each cell is a histogram of 8 orientation bins relative to dominant orientation



compute descriptor, relative
to dominant orientation



128 dim
descriptor

each descriptor has:

$$P_i = (x_i, y_i, \rho_i, \vartheta_i) \quad \text{and} \quad f_i \dots 128 \text{ dim vector}$$

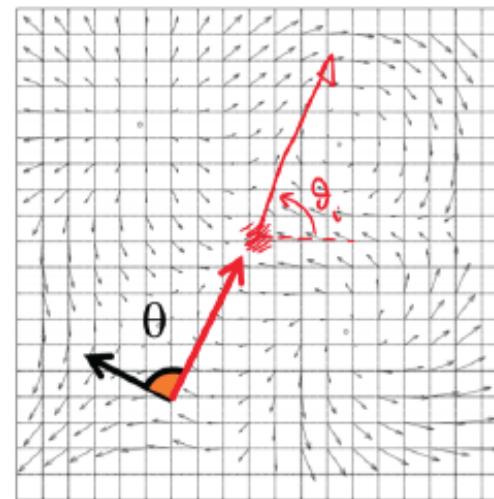
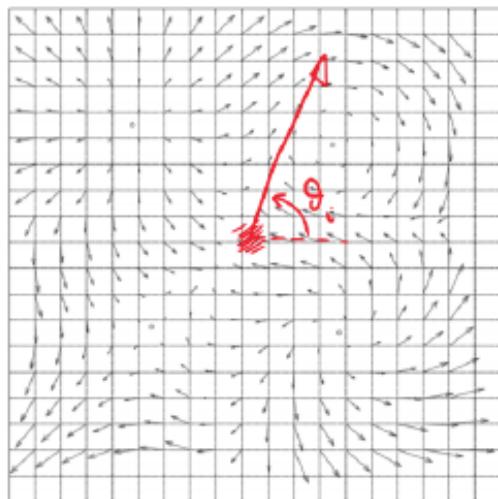
↑ ↑ ↑ ↑
location scale orientation feature vector

[Adopted from: F. Flores-Mangas]

SIFT Descriptor: Computing the Feature Vector

- Compute the orientations **relative** to the **dominant orientation**

16×16 patch
centered in (x_i, y_i)

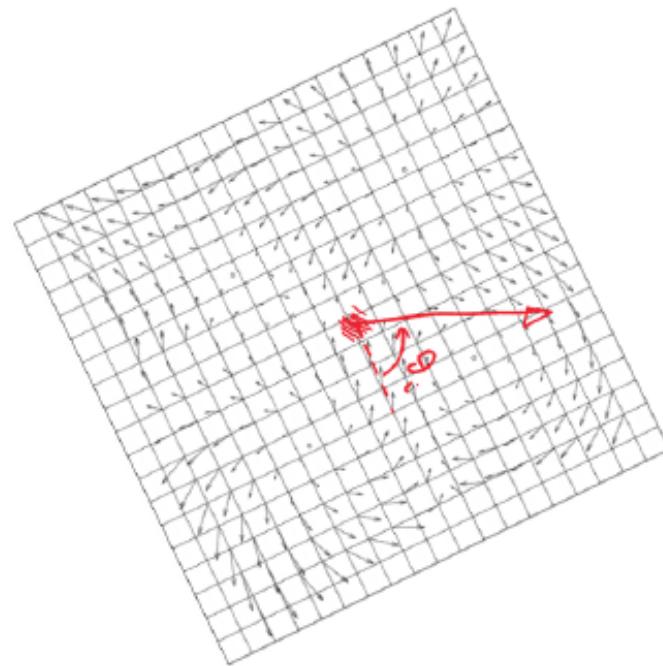
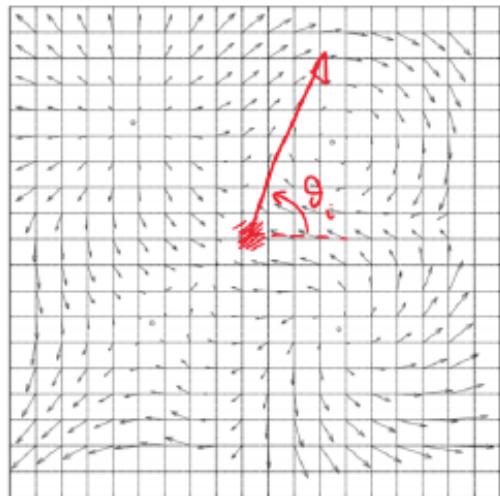


[Adopted from: F. Flores-Mangas]

SIFT Descriptor: Computing the Feature Vector

- Compute the orientations **relative** to the **dominant orientation**

16×16 patch
centered in (x_i, y_i)

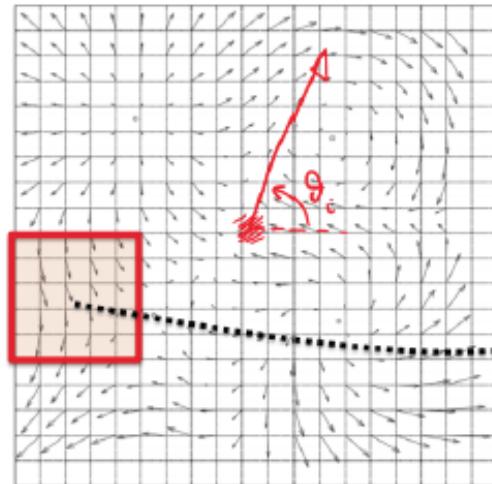


[Adopted from: F. Flores-Mangas]

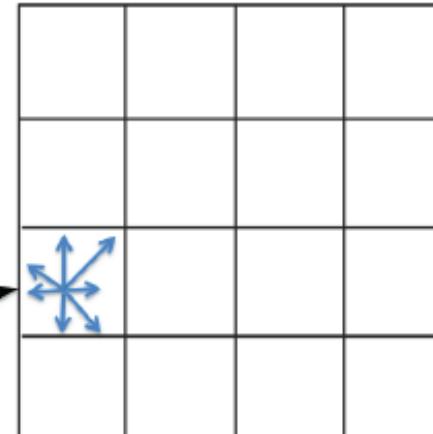
SIFT Descriptor: Computing the Feature Vector

- Compute the orientations **relative to the dominant orientation**
- Form a 4×4 grid. For each grid cell compute a histogram of orientations for 8 orientation bins spaced apart by 45°

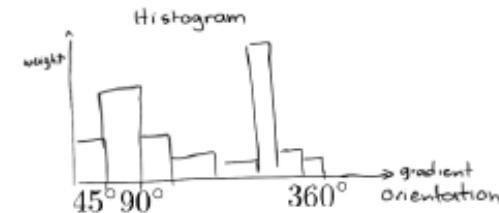
16×16 patch
centered in (x_i, y_i)



SIFT descriptor



compute histogram of orientations
this time 8 bins spaced by 45°

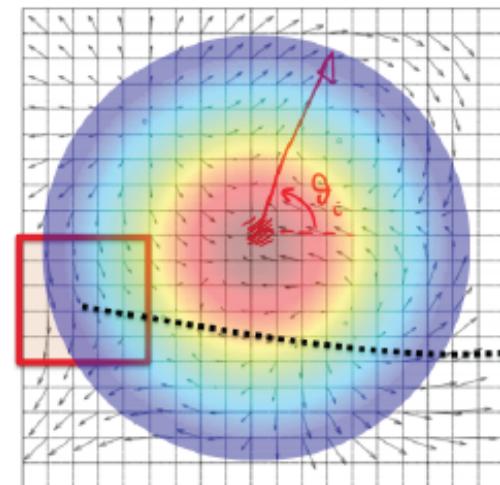


[Adopted from: F. Flores-Mangas]

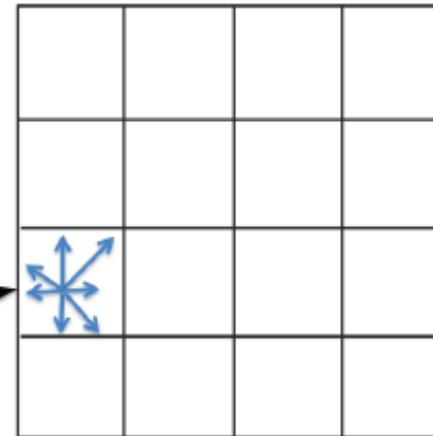
SIFT Descriptor: Computing the Feature Vector

- Compute the orientations **relative** to the **dominant orientation**
- Form a 4×4 grid. For each grid cell compute a histogram of orientations for 8 orientation bins spaced apart by 45°

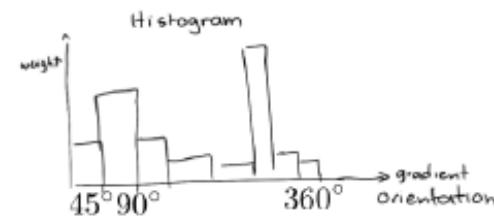
16×16 patch
centered in (x_i, y_i)



SIFT descriptor



again weigh contributions
this time: $|\nabla I(x, y)| \cdot G_{0.5\rho}$

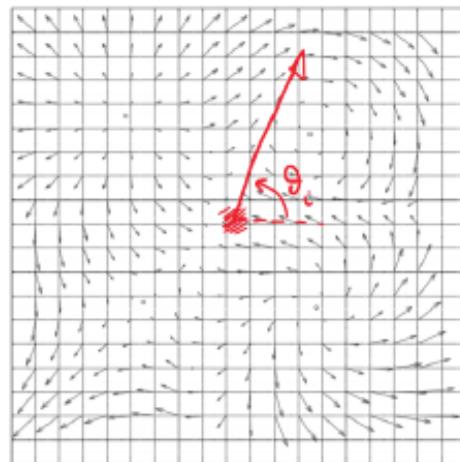


[Adopted from: F. Flores-Mangas]

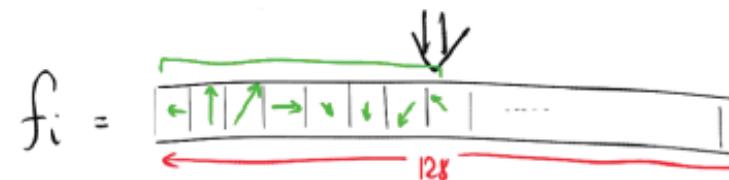
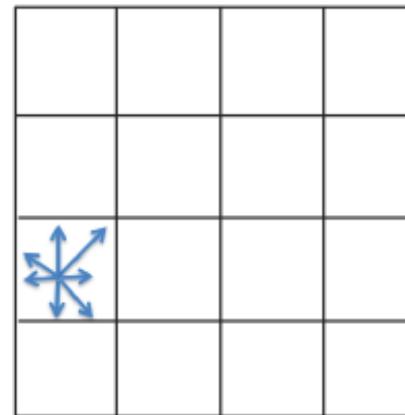
SIFT Descriptor: Computing the Feature Vector

- Compute the orientations **relative** to the **dominant orientation**
- Form a 4×4 grid. For each grid cell compute a histogram of orientations for 8 orientation bins spaced apart by 45°
- Form the 128 dimensional feature vector

16×16 patch
centered in (x_i, y_i)



SIFT descriptor



[Adopted from: F. Flores-Mangas]

SIFT Descriptor: Post Processing

- The resulting 128 non-negative vector is the raw version of the descriptor
- To reduce effects of contrast or gain:
 - normalize the vector to be unit norm : $f_i = f_i / \|f\|$
- To reduce effects of illumination variations:
 - clip values to 0.2, and renormalize to unit norm

SIFT Properties:

Invariant to:

Scale

Rotation

Partially invariant to:

Illumination changes

Camera viewpoint (up to 60 degrees out of plane rotation)

Occlusion, clutter

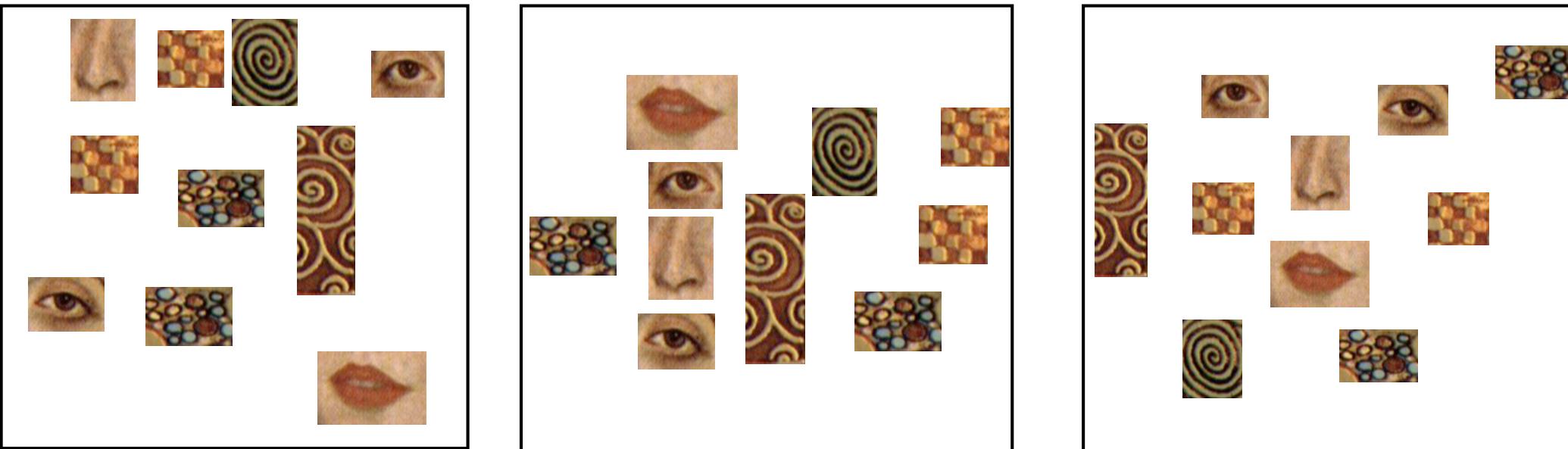
Fast and efficient

Code available

PCA-SIFT

- The dimension of the descriptor is large
- Can use PCA to reduce the dimension ~10

Problem with bag-of-words



All have equal probability for bag-of-words methods

Location information is important

Bag of Words and Spatial Information

A bag of words throws away spatial relationships between features.

Middle ground:

Visual “phrases”: frequently co-occurring words

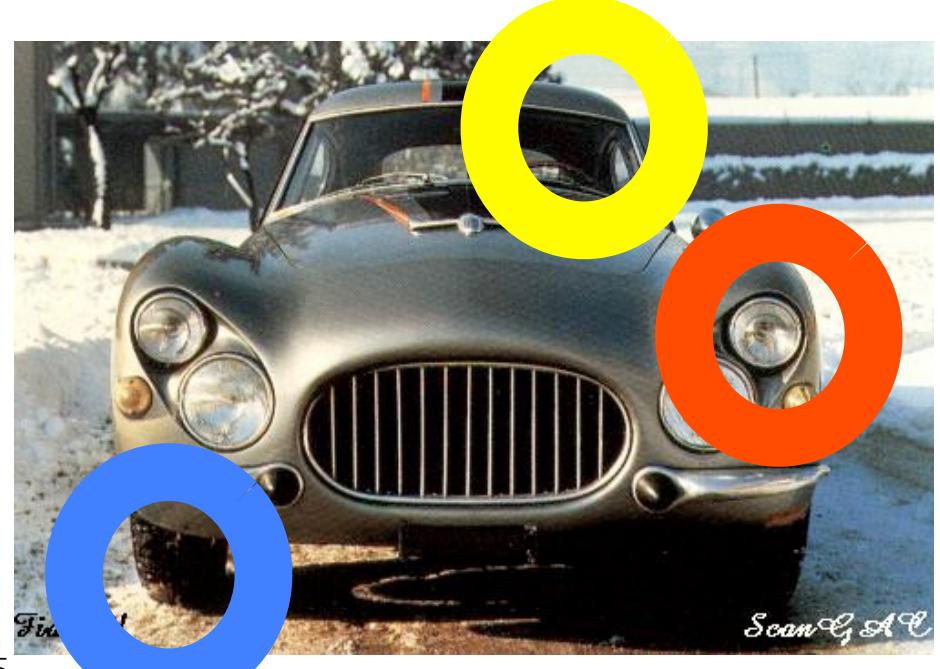
Semi-local features: describe configuration, neighborhood

Let position be part of each feature

Count bags of words only within sub-grids of an image

After matching, verify spatial consistency

Parts & Structure



Implicit Shape Model

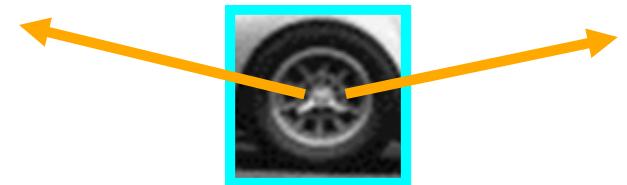
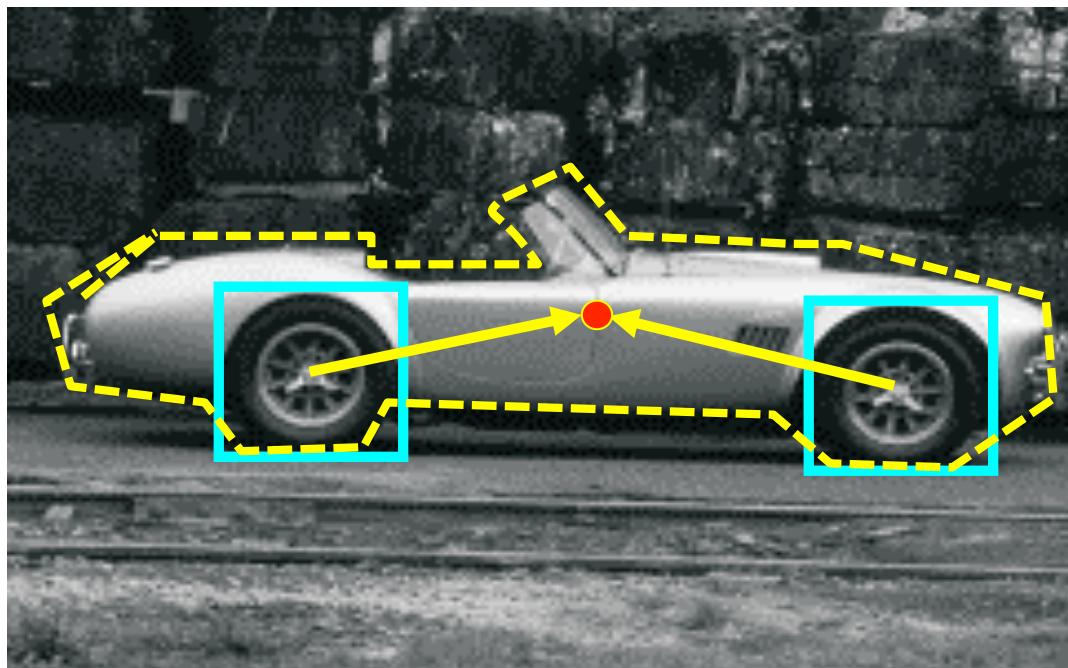
Use Hough Space Voting to find object:

- Learn spatial distributions of the words wrt to a “reference point”

- Use HT to vote for reference points in the test image

Implicit shape models

- Visual vocabulary is used to index votes for object position
 - [a visual word = “part”]



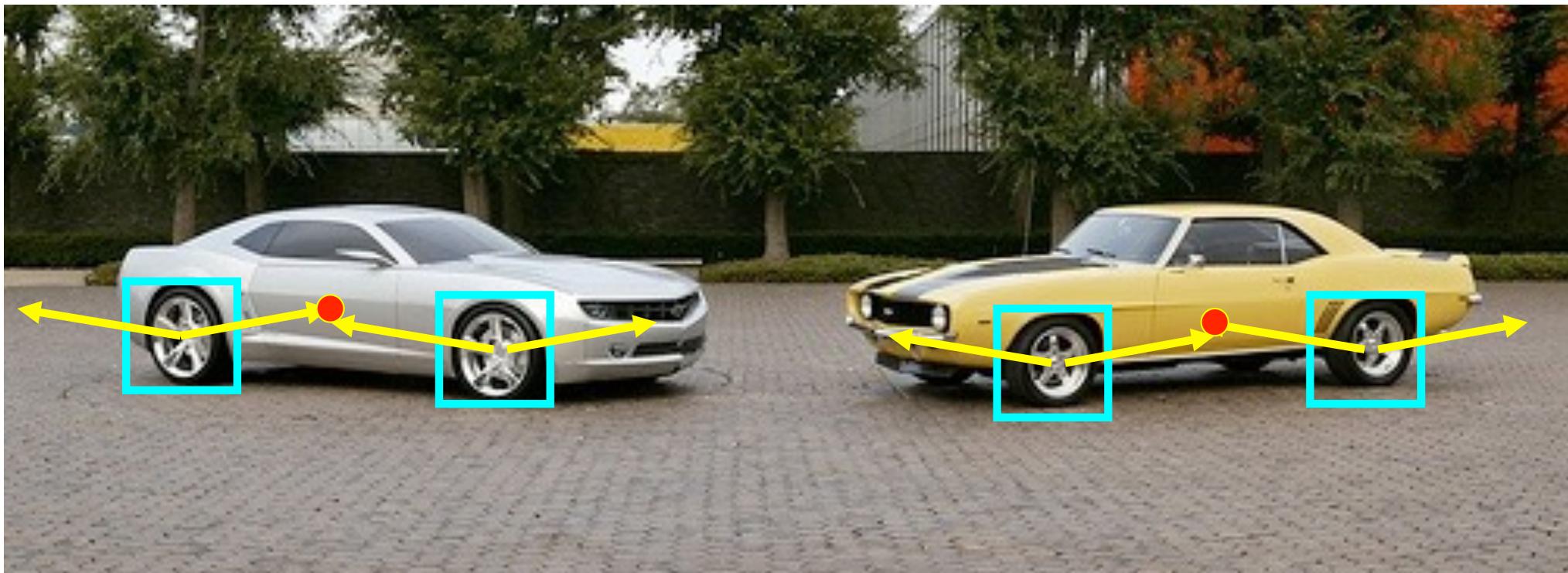
visual codeword with
displacement vectors

training image annotated with object localization info

B. Leibe, A. Leonardis, and B. Schiele, [Combined Object Categorization and Segmentation with an Implicit Shape Model](#), ECCV Workshop on Statistical Learning in Computer Vision 2004

Implicit shape models

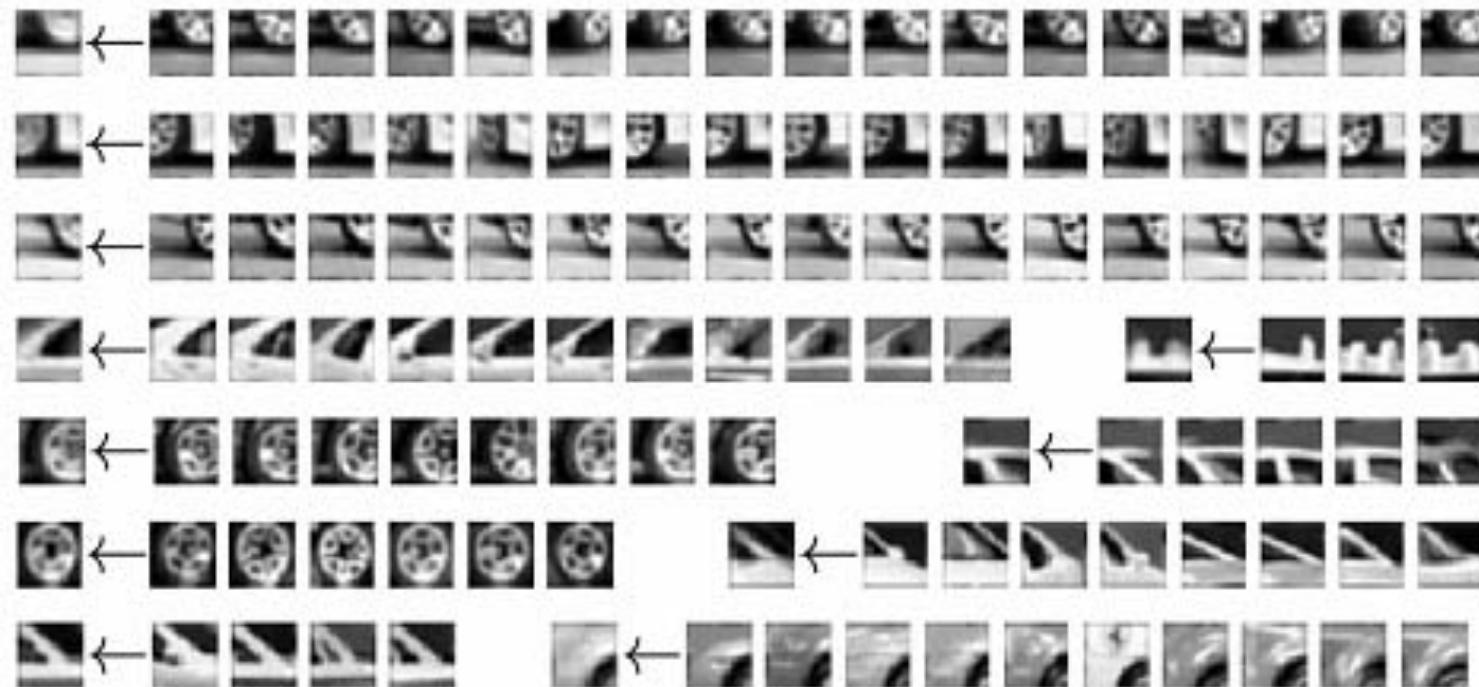
- Visual vocabulary is used to index votes for object position
 - [a visual word = “part”]



test image

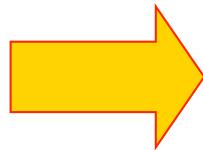
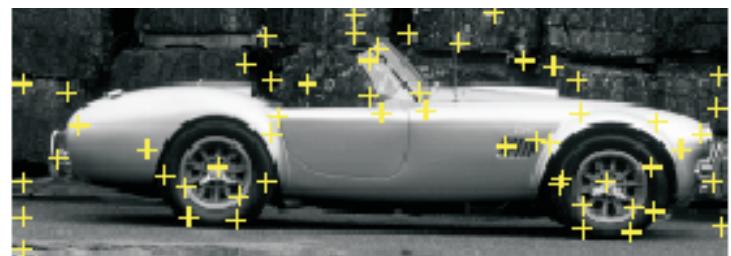
Implicit shape models: Training

1. Build vocabulary of patches around extracted interest points using clustering



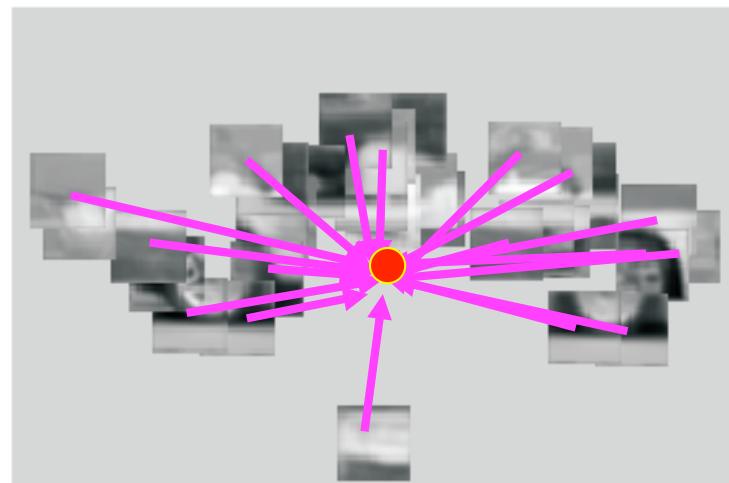
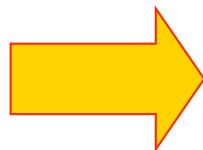
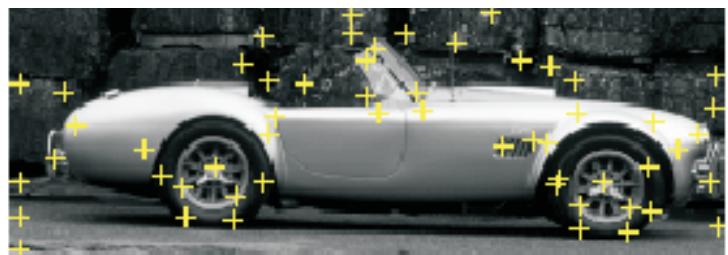
Implicit shape models: Training

1. Build vocabulary of patches around extracted interest points using clustering
2. Map the patch around each interest point to closest word



Implicit shape models: Training

1. Build vocabulary of patches around extracted interest points using clustering
2. Map the patch around each interest point to closest word
3. For each word, store all positions it was found, relative to object center

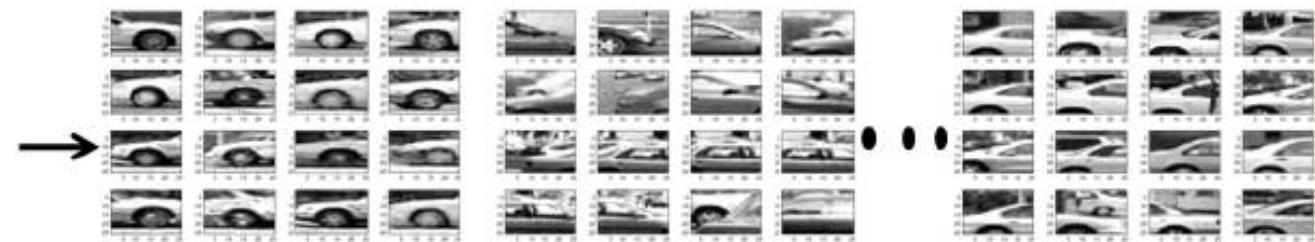


Training

Training stage:



Training examples



Discovered local parts:
visual words

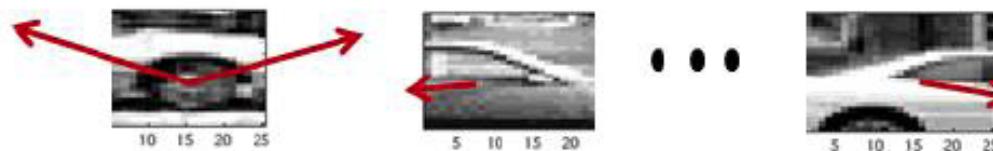


Table of displacement vectors

Implicit shape models: Testing

1. Given new test image, extract patches, match to vocabulary words
2. Cast votes for possible positions of object center
3. Search for maxima in voting space
4. (Extract weighted segmentation mask based on stored masks for the codebook occurrences)

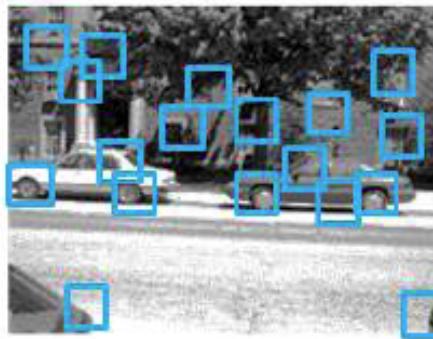
What is the dimension of the Hough space?

Detecting

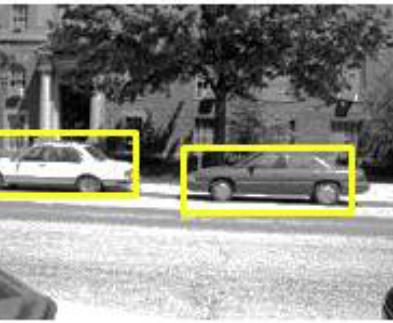
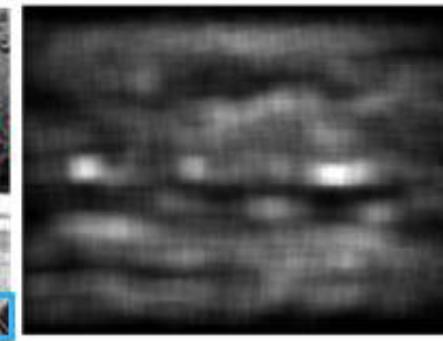
Detection stage:



Novel image



Voting (GHT)



Predicted object
locations

Detection Results

Qualitative Performance

Recognizes different kinds of objects

Robust to clutter, occlusion, noise, low contrast



Example: Results on Cows



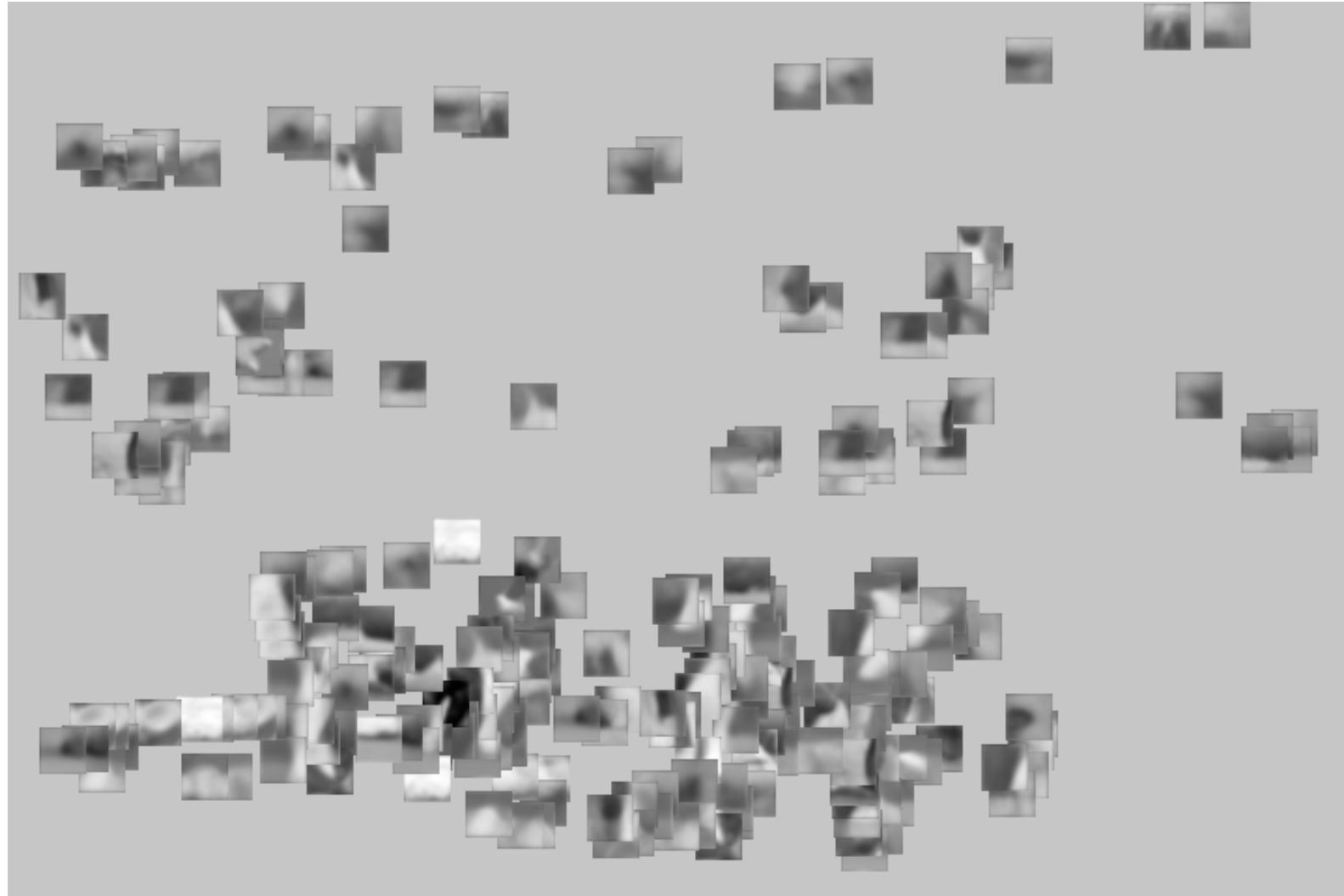
Original image

Example: Results on Cows



Interest points

Example: Results on Cows



Matched patches points

Example: Results on Cows



Matched

Votes

99

Example: Results on Cows



1st hypothesis

Example: Results on Cows



2nd hypothesis

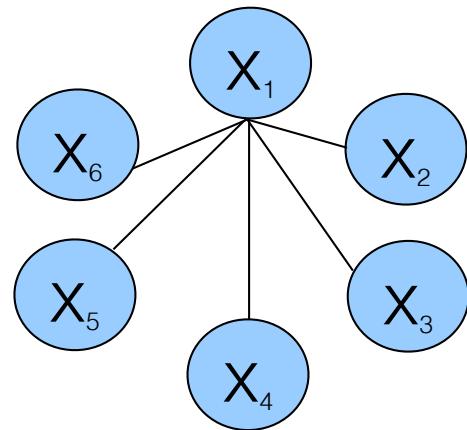
Example: Results on Cows



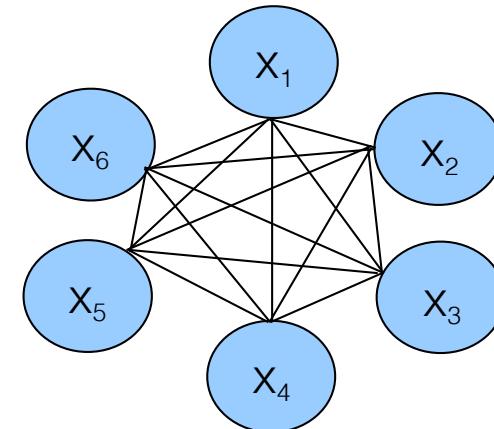
3rd hypothesis

Shape representation in part-based models

“Star” shape model



Fully connected constellation model



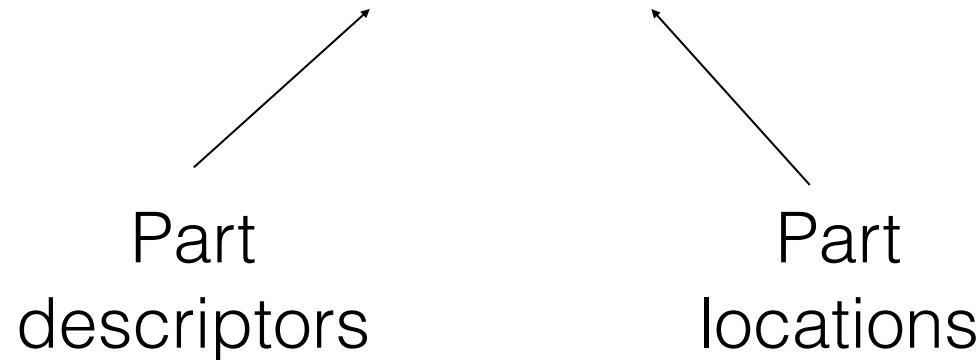
- e.g. implicit shape model
- Parts mutually independent

- e.g. Constellation Model
- Parts fully connected

N image features, P parts in the model

Probabilistic constellation model

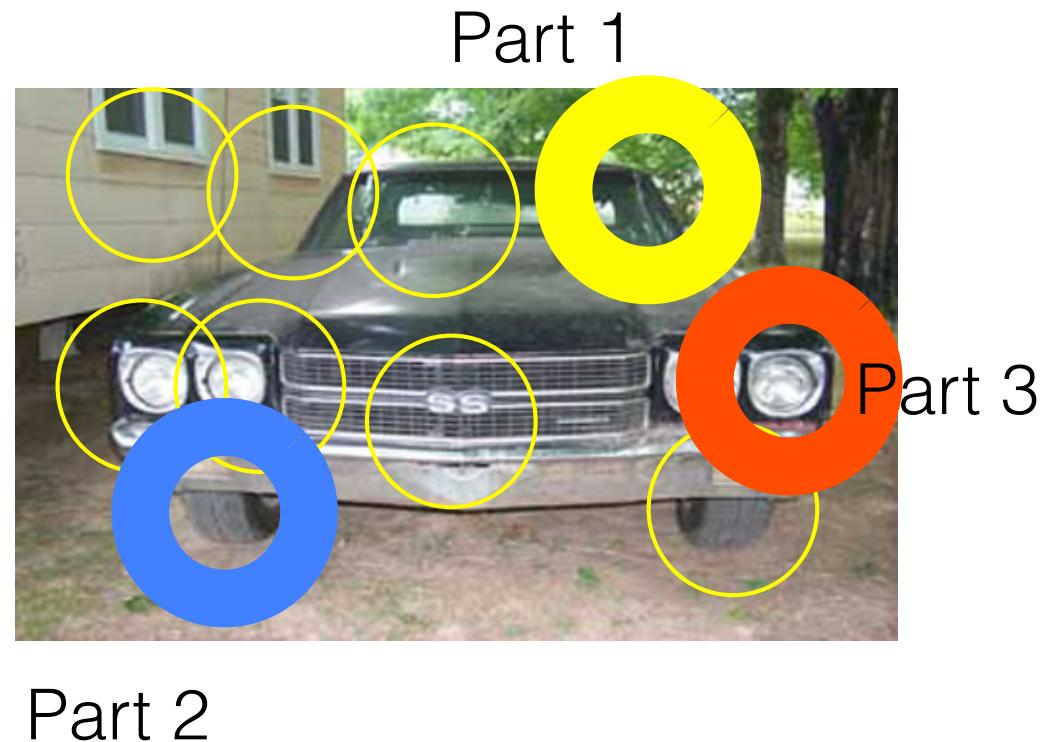
$$P(\text{image} \mid \text{object}) = P(\text{appearance}, \text{shape} \mid \text{object})$$



Candidate parts

Probabilistic constellation model

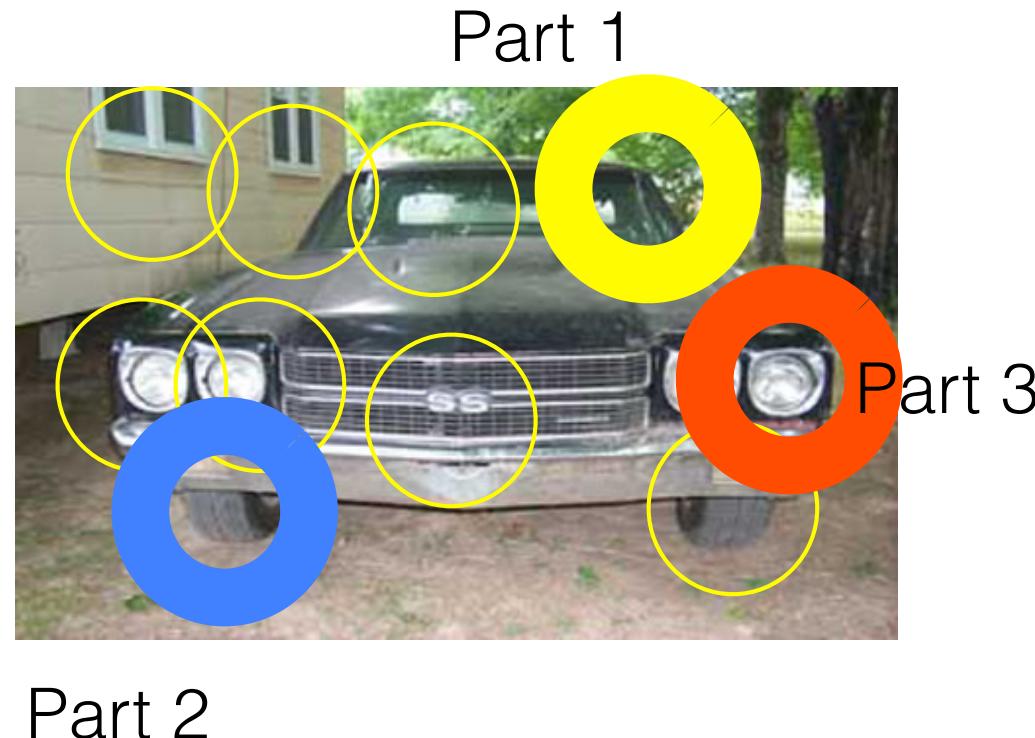
$$P(\text{image} \mid \text{object}) = P(\text{appearance}, \text{shape} \mid \text{object})$$



Probabilistic constellation model

$$\begin{aligned} P(\text{image} \mid \text{object}) &= P(\text{appearance}, \text{shape} \mid \text{object}) \\ &= \max_h P(\text{appearance} \mid h, \text{object}) p(\text{shape} \mid h, \text{object}) p(h \mid \text{object}) \end{aligned}$$

h : assignment of features to parts



Example results from constellation model: data from four categories

Faces



Motorbikes



Airplanes



Spotted cats



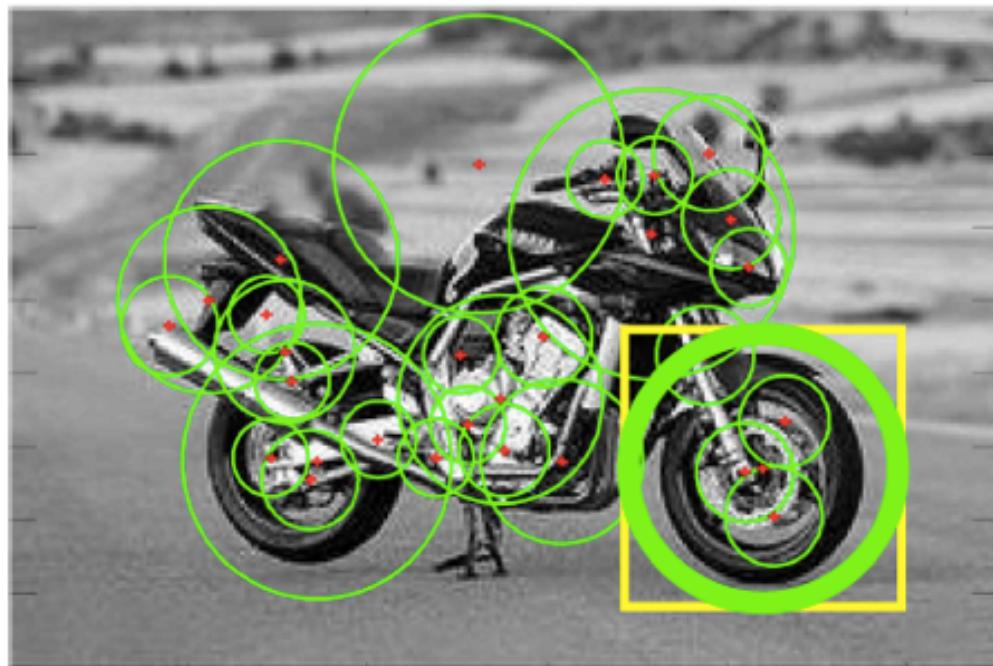
Weakly Supervised Class Learning

Train class without segmentation

Training images have the object of interest (once)
but do not know where or the scale



Semi-supervised Training (Fergus et al '03)



Appearance

- Find regions within image
- Use salient region operator
(Kadir & Brady 01)

Location

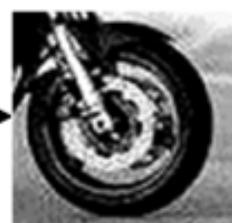
(x,y) coords. of region centre

Scale

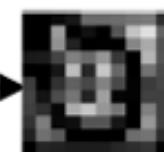
Radius of region (pixels)



Normalize



11x11 patch



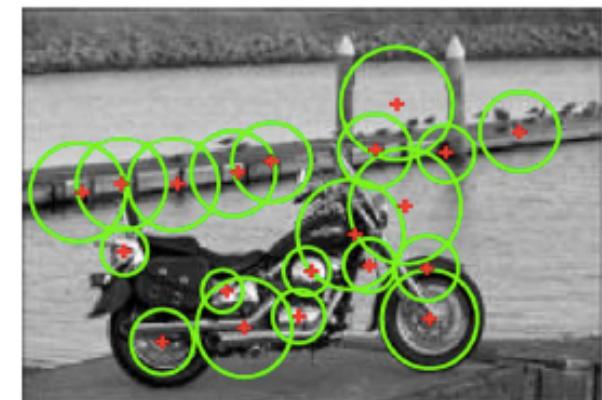
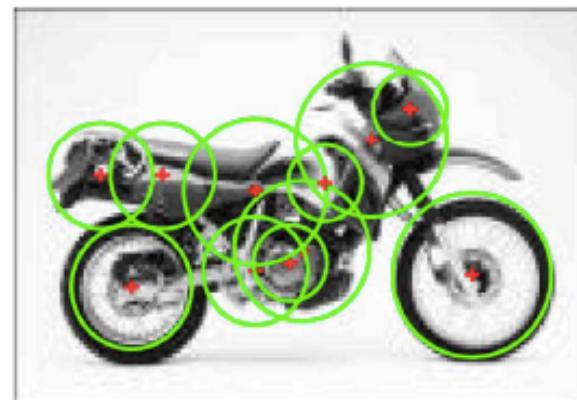
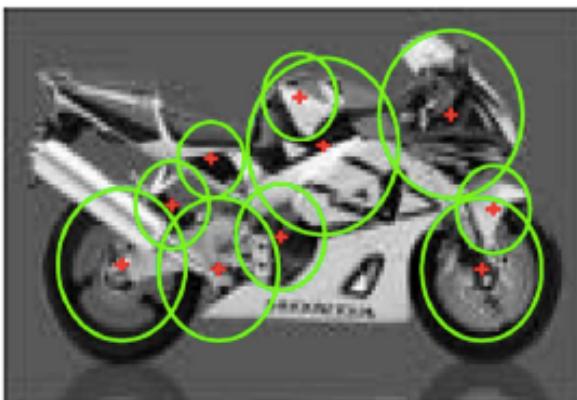
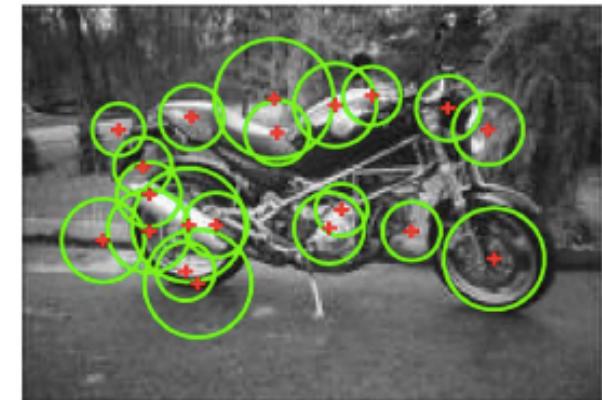
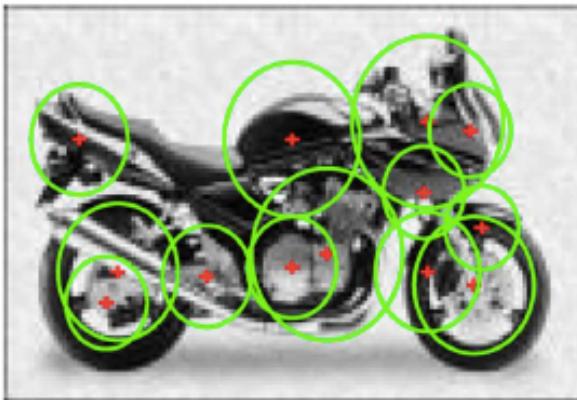
Projection onto
PCA basis

$$\begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_{15} \end{pmatrix}$$

Gives representation of appearance in low-dimensional vector space

Motorcycle Example

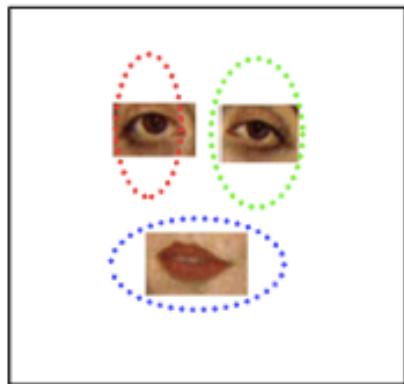
- Kadir & Brady saliency region detector



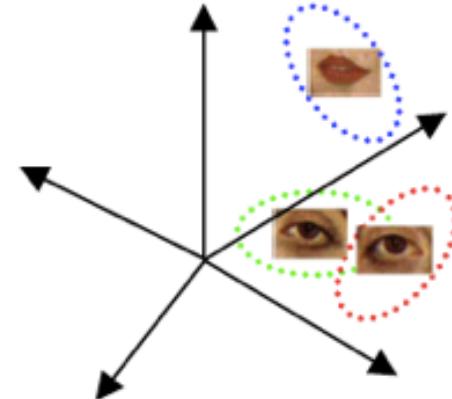
Generative Model

Foreground model

Gaussian shape pdf



Gaussian part appearance pdf



based on Burl, Weber et al. [ECCV '98, '00]

Gaussian

relative scale pdf

$\log(\text{scale})$

Prob. of detection

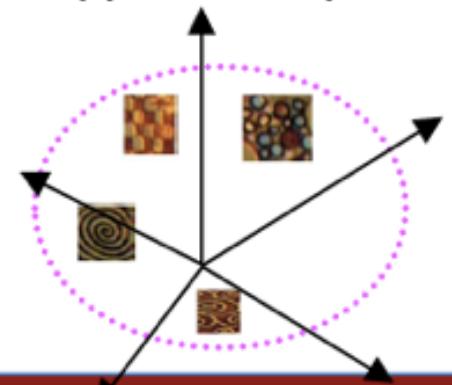


Clutter model

Uniform shape pdf



Gaussian background
appearance pdf



Uniform

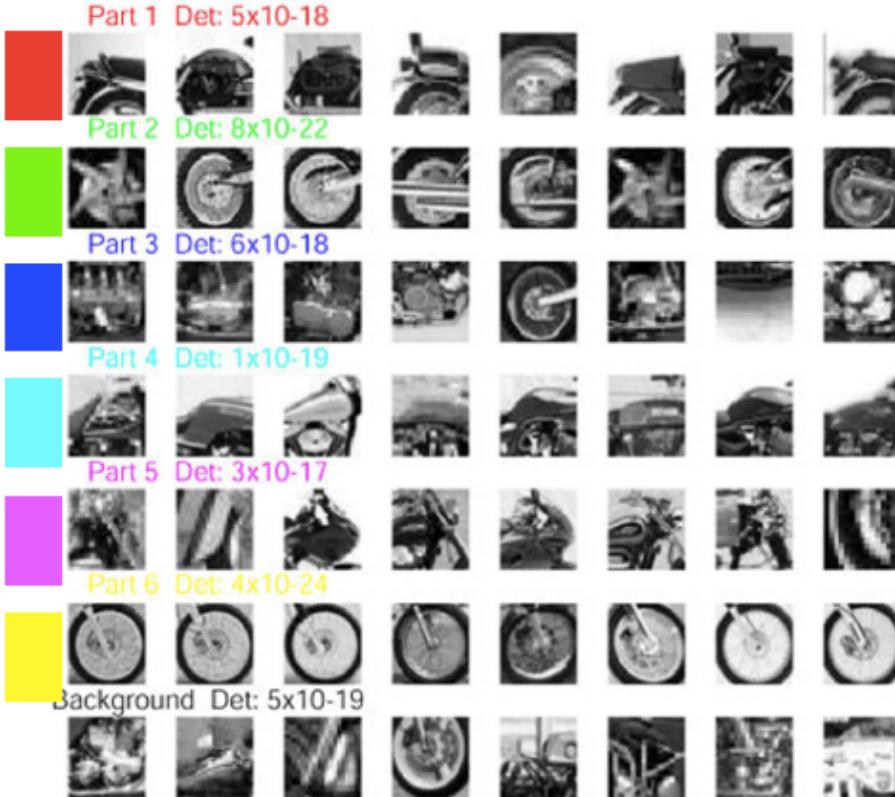
relative scale pdf

$\log(\text{scale})$

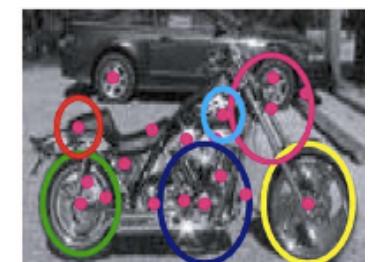
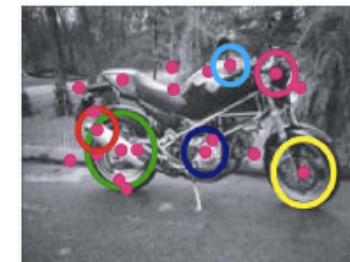
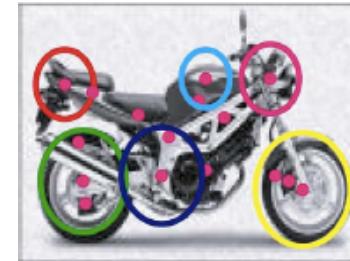
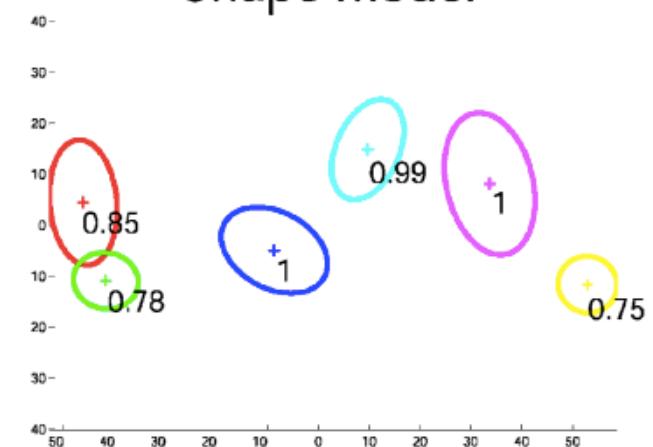
Poisson pdf on #
detections

Motorbikes

Samples from appearance model

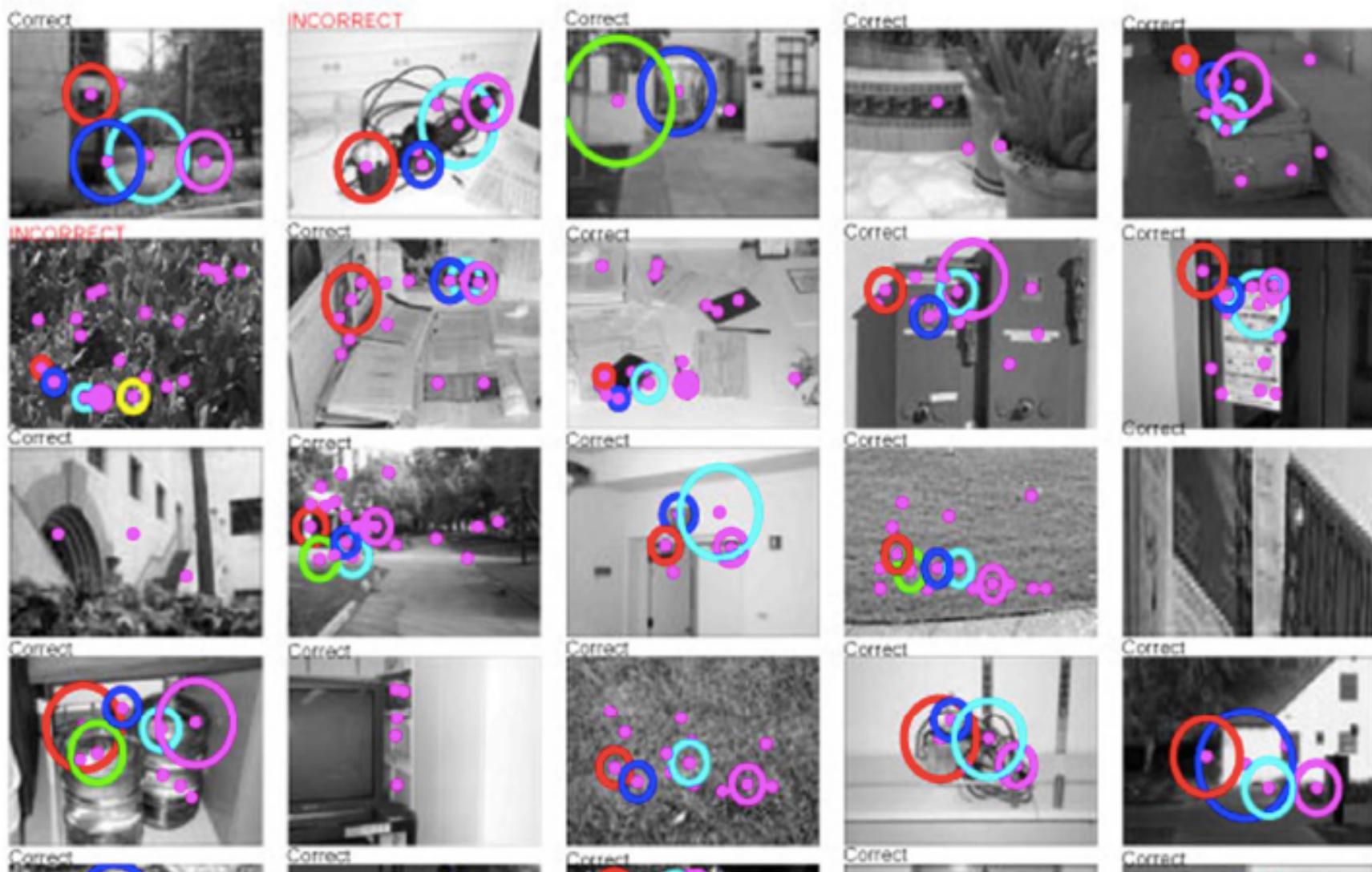


Shape model

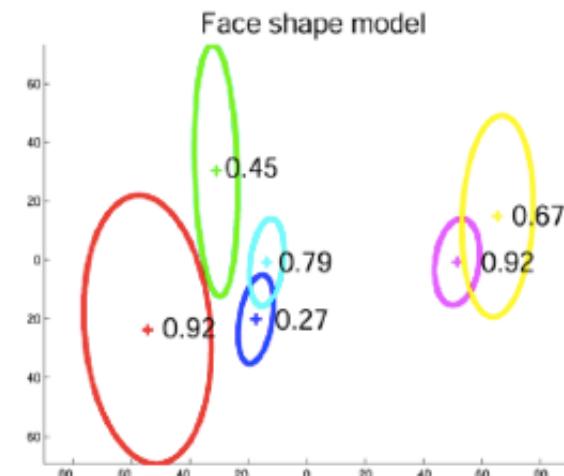
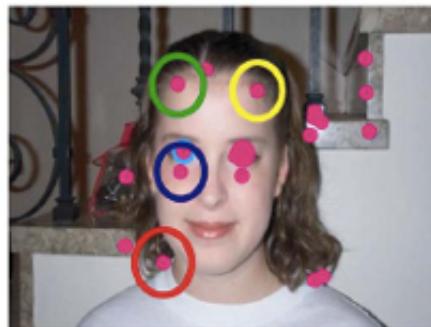


Test images: size of circles indicates score of hypothesis

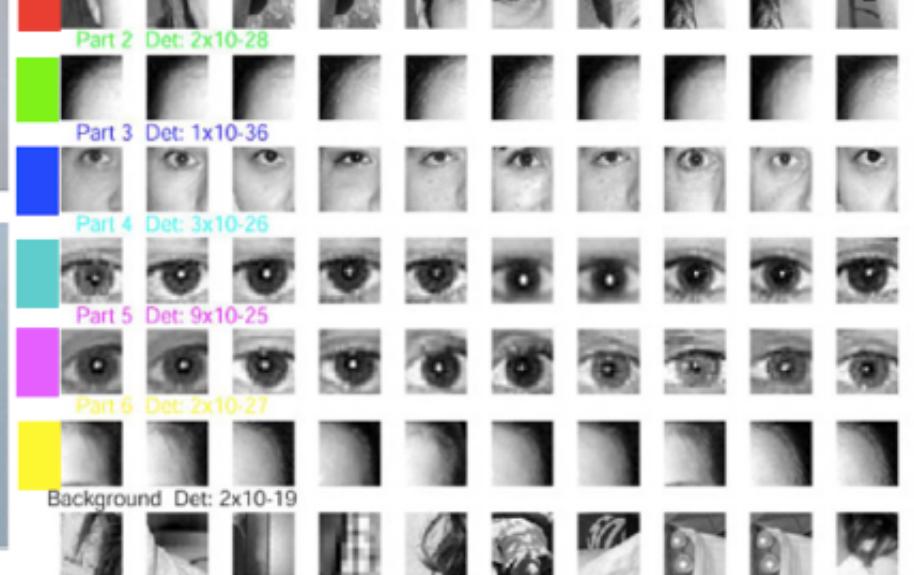
Motorbike Model on Background Images



Frontal Faces



Part 1 Det: 5x10-21

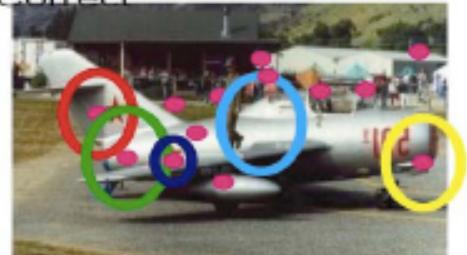


Airplanes

INCORRECT



Correct



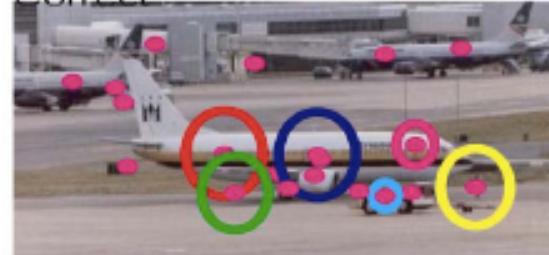
Correct



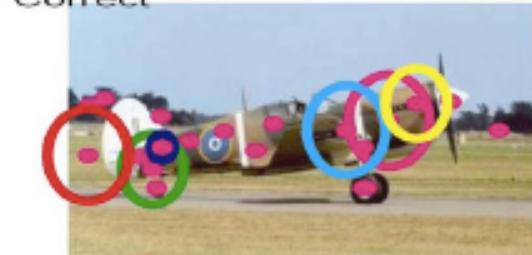
Correct



Correct



Correct



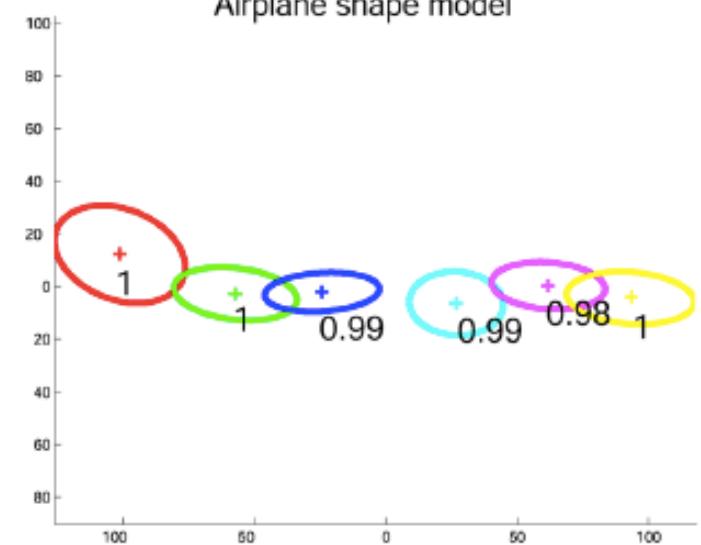
Correct



Correct



Airplane shape model



Part 1 Det: 3x10-19



Part 2 Det: 9x10-22



Part 3 Det: 1x10-23



Part 4 Det: 2x10-22



Part 5 Det: 7x10-24



Part 6 Det: 5x10-22



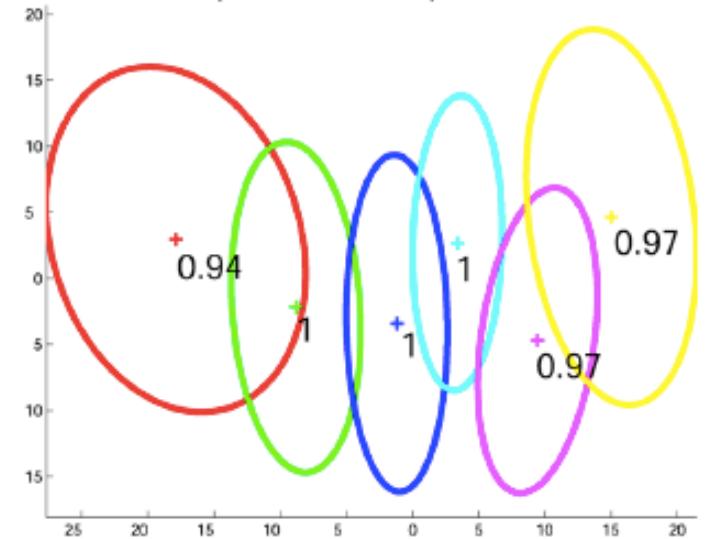
Background Det: 1x10-20



Spotted Cats



Spotted cat shape model



Part 1 Det: 8x10-22



Part 2 Det: 2x10-22



Part 3 Det: 5x10-22



Part 4 Det: 2x10-22



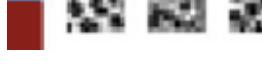
Part 5 Det: 1x10-22



Part 6 Det: 4x10-21



Background Det: 2x10-18



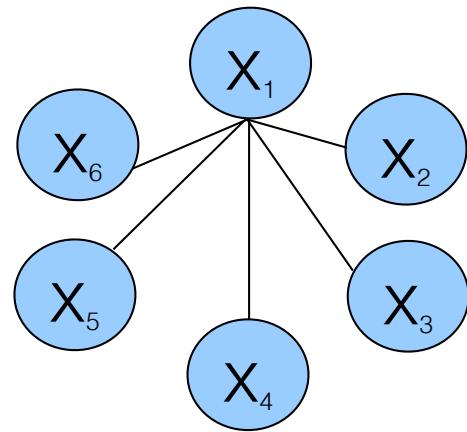
Comparison



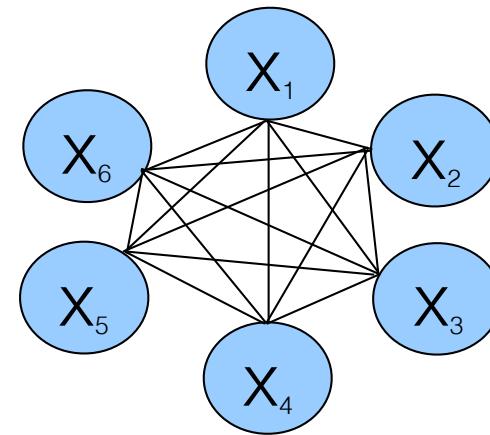
class	bag of features	bag of features	Part-based model
	Zhang et al. (2005)	Willamowski et al. (2004)	Fergus et al. (2003)
airplanes	98.8	97.1	90.2
cars (rear)	98.3	98.6	90.3
cars (side)	95.0	87.3	88.5
faces	100	99.3	96.4
motorbikes	98.5	98.0	92.5
spotted cats	97.0	—	90.0

Shape representation in part-based models

“Star” shape model



Fully connected constellation model



- e.g. implicit shape model
- Parts mutually independent
- Recognition complexity: $O(NP)$
- Method: Gen. Hough Transform

- e.g. Constellation Model
- Parts fully connected
- Recognition complexity: $O(N^P)$
- Method: Exhaustive search

N image features, P parts in the model

Summary:

part-based and local feature models for generic object recognition

Histograms of visual words to capture global or local layout in the bag-of-words framework

Powerful in practice for image recognition

Part-based models encode category's part appearance together with 2d layout and allow detection within cluttered image

“**implicit shape model**”: shape based on layout of all parts relative to a reference part; Generalized Hough for detection

“**constellation model**”: explicitly model mutual spatial layout between all pairs of parts; exhaustive search for best fit of features to parts